SUPERVISED EXTRACTION OF FACE SUBSPACES BASED ON MULTIMODAL DISCRIMINANT ANALYSIS

LI JIANRAN

NATIONAL UNIVERSITY OF SINGAPORE

2009

SUPERVISED EXTRACTION OF FACE SUBSPACES BASED ON MULTIMODAL DISCRIMINANT ANALYSIS

LI JIANRAN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE THE DEGREE OF

Master of Science

in

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

SINGAPORE, 2009

To my parents and grandparents

Acknowledgements

I would like to thank

My advisor Dr. Terence Sim for his invaluable guidance, support and understanding. He introduced me to the interesting problems of face recognition, FFKT and MMDA. His enthusiasm on the topic and positive outlook has always inspired me.

Dr. Leow Wee Kheng and Dr. Michael Brown, for wonderful suggestions and discussions.

My senior, Dr. Zhang Sheng, for his inspirational works which have provided the foundation for this thesis, for the valuable discussions and for his encouragement.

My colleagues at Computer Vision Lab for their friendship, help and support, especially Zhang Xiaopeng, Ye Ning, Guo Dong, Li Hao, Zhuo Shaojie and Lu Zheng. They have made my research experience a pleasant and enriching journey.

My dear friends, Zhang Xiaoming and Wong Wei Pin, for always being there for me, whenever I need them and even when I do not.

My beloved parents and grandparents, for giving me strength and courage throughout my studies and leading to the completion of this thesis.

Abstract

Face image appearance may change due to a variety of factors (or modes) such as the person's identity, lighting condition and expression. We propose a method for representing these appearance changes as a mixture of different modes in different subspaces. These subspaces are simultaneously extracted in the following manner: we first transform the data to the whitened space and then perform Fisher Discriminant Analysis (FDA) to find mutually orthogonal discriminant subspaces for different modes based on their respective labeling information. The proposed method could be used for dimension reduction and face recognition. To validate the effectiveness of the method, we have tested it on the Multi-PIE database. Experiment results of dimension reduction show satisfactory visual quality and those of face recognition show superior performance compared to PCA and LDA.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Overview

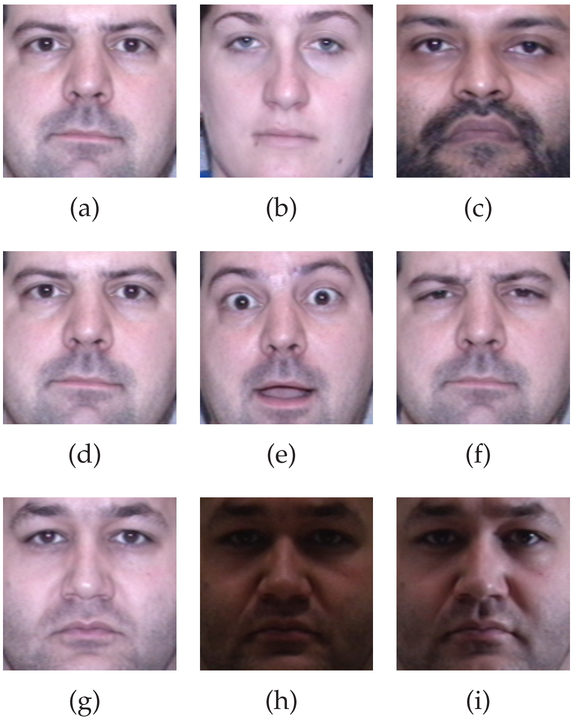


Figure 1.1: Illustration of different modes of facial image variation. (a) (b) (c) Variation of person identity. (d) (e) (f) Variation of expression. (g) (h) (i) Variation of illumination condition.

Face images are generated by the interaction of multiple factors (or modes) related to the person's identity, illumination condition, expression, and viewpoint, etc. Figure 1.1 shows some examples of face images with multiple variation modes.

In real-life applications, we are often more interested in one particular factor than others. For example, a face recognition system needs to recognize a person's identity regardless of his/her expression or pose; a lip-reader will concentrate only on the expression; in face image synthesis, we wish to be able to modify certain factors of the face images (e.g. expression, lighting) while keeping the others invariant (e.g. identity). The understanding of how these different factors affect the appearance of a face image, and the relationships between them, is thus critical. Particularly, we are interested in (but not restricted to) the following issues:

- Given a face image, can we identify the components of the image corresponding to different variation modes?
- Can we extract these components separately?
- Are these components independent among them?

These issues are clearly related but also have subtle differences. The first one is mainly concerned with understanding the face images, e.g. the main causes for appearance variation and the corresponding components embedded in the face images. A typical application scenario is face recognition, where we are interested in identifying the component of a face image which reflects the person's identity. The second issue goes one level higher: we are not only interested in identifying these variation components, we wish to extract them separately; this implicitly requires that these components have no overlapping regions and are indeed separable. The third issue imposes more constraints than the previous one: if the different variation components can indeed be extracted separately, we ask whether these components are independent among them. We are interested in this issue because only when this condition is satisfied can we manipulate the different components separately without affecting the others. For example, in face image synthesis, we may wish to be able to manipulate the expression and illumination condition of a face image without changing the person's identity.

To date, there is no consensus regarding answers to the above questions. The main reason is that there is no general understanding of how the face space[1] looks like and most of the time, the above issues have been approached in an application-oriented manner. Numerous algorithms have been proposed to better understand the variation of the face images subject to various factors and their effectiveness varies in different application scenarios. Eigenface [27, 28] finds *principal components* (PCs) with different weights (or energy) to represent face images efficiently. The PCs are *uncorrelated*, but there is no information regarding the variation factor associated with each PC, thus none of the issues above are answered. Fisherface [2] extracts *discriminant subspaces* for face recognition; it only considers the variation factor of person identity, and aims to project all other types of variation out of the *discriminant subspaces*, thus providing a partial answer to the first two issues above. Costen et al. [8] attempt to find functional face subspaces, each containing information regarding a certain variation factor, but there are overlapping regions among these face subspaces, thus fail to address the second and third issues. TensorFace [29, 30, 31, 32] successfully extracts the face subspaces of the different variation factors using multi-linear algebra, but avoids to discuss the inherent separability and independence of these factors explicitly, thus the third issue remains.

1.2 Our Approach

Figure 1.2 illustrates the analysis framework of the thesis. Firstly, we whiten the data set to decorrelate the axes of the vectors represented by the face images. Secondly, analysis is carried out in the whitened space[2], and subspaces which purely contains information of a given mode are extracted based on labeling information of the specific mode. These extracted subspaces which contain information of different variation modes constitute the **Subspace of interest** in the whitened data space. Particularly, face recognition can be carried out by studying projections of the images onto the subspace which contains information regarding the people's

[1] The space spanned by all the face images with all possible variation modes, including the person's identity, expression, illumination and pose. Attempts of studying this space can be found at [25, 34].

[2] As will be defined in Section 3.2.

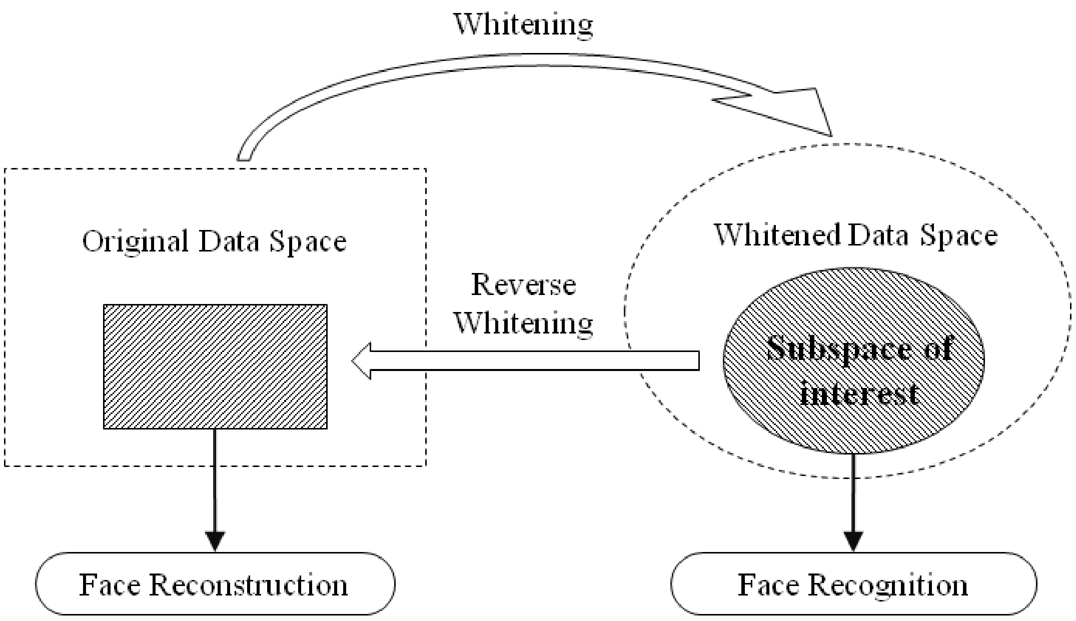


Figure 1.2: Illustration of the analysis framework of the thesis.

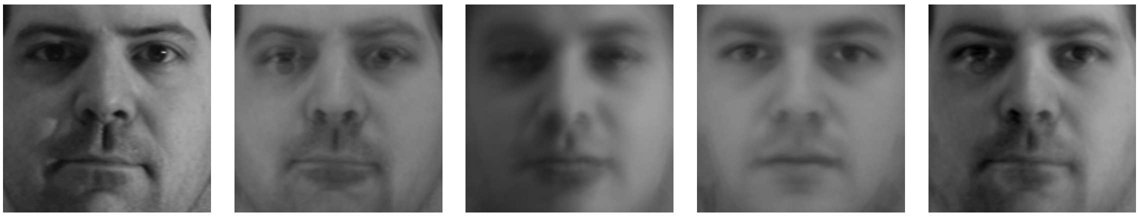
identity. A dimension reduction procedure can be performed by projecting the face images onto the *Subspace of interest* and reconstructing them using only the basis of that subspace. Finally, the *Subspace of interest* in the whitened data space is reverse-whitened to be mapped back to the original data space; similarly for the reconstructed face images. The reverse-whitened face images in the original data space can then be used as the reconstructed face images. Figure 1.3 shows an example of a reconstructed image using (reverse-whitened) orthogonal variation components in the whitened data space.

The whitened space is decomposed[3] into orthogonal subspaces in the following form:

$$\mathscr{D} = \mathscr{U}_{people} \oplus \mathscr{U}_{illumination} \oplus \mathscr{U}_{expression} \oplus \cdots \oplus \mathscr{R} \tag{1.1}$$

where $\mathscr{D}$ denotes the whitened data space, $\mathscr{U}$ denotes the subspaces of different

[3] As we shall see in Section 4.3, the decomposition holds under certain assumptions, which are usually satisfied in applications related to face images.



(a) Original Image (b) Person identity component (c) Illumination component (d) Expression component (e) Reconstructed image using the three components

Figure 1.3: Illustration of face image reconstruction using our method.

modes, $\mathscr{R}$ denotes the subspace containing information which have not yet been captured, and $\oplus$ denotes an orthogonal direct sum[4] of the subspaces (thus there is no overlapping region and they are *uncorrelated*). The orthogonal direct sum of different $\mathscr{U}$ constitutes, in the fact, the *Subspace of interest*. The orthogonal decomposition in the whitened data space corresponds to in fact an oblique decomposition in the original data space, as demonstrated in Section 3.2.5. This implies that even if the components corresponding to different variation modes are inherently correlated, we could still find an *uncorrelated embedding* in the whitened space.

Our method, in fact, combines ideas from several existing approaches. Firstly, our approach is based on the *Fisher Discriminant Analysis* (FDA), and the extracted subspace of a particular mode is the most discriminating subspace for the mode. Secondly, our analysis is carried out in the whitened data space , in which the axes are uncorrelated (as in *PCA*) and normalized in inverse proportion to their covariance (as in *AAM functional face subspaces*). This data space has shown desirable properties in face recognition as studied in [36]. Finally, our method is similar to *Tensorfaces* in that we achieve a clean decomposition of face images into several subspaces characterized by the corresponding variation mode (with an additional residual term). Our method constructs the mutually orthogonal subspaces in the whitened space, regardless of their correlation in the original subspace (where most probably they are highly correlated), unlike *TensorFaces* which implicitly embed the assumption of uncorrelatedness in their representation.

[4] If a space S is written as the direct sum of subspaces $S_1, \ldots, S_n$, then we have $S = S_1 + \cdots + S_2$, and $S_i \bigcap S_j = \emptyset$ for $i \neq j$. The direct sum is orthogonal if $S_i \perp S_j$ for $i \neq j$.

1.3 Contributions and Applications

The thesis is built upon, consolidates and reformulates previous (published and unpublished) works in [35, 36, 37, 38, 39] and proposes an analysis framework for the study of face image variations. More specifically, our approach can extract different variation components of face images separately, which improves understanding of face images and allows easy manipulation of image appearance.

As indicated in Figure 1.1, our method could easily be applied to face recognition, dimension reduction and face image synthesis[5].

1.4 Thesis Outline

The remainder of the thesis is organized as follows. Chapter 2 is a literature survey on related works in the fields of face recognition and dimension reduction. Chapter 3 provides the mathematical background of some terms and algorithms which will be used frequently in later chapters. Chapter 4 lays the theoretical foundation for the development of our method, which is critical for the understanding for later chapters. Chapter 5 is the core of the theoretical development of our method. Chapter 6 demonstrates the strengths of our method in two domains: face image decomposition and face recognition. Finally, Chapter 7 concludes the thesis.

1.5 Notations

For the convenience of presentation, we shall use the same notation system in this thesis. Scalar variables will be denoted using uppercase or lowercase italicized letters, such as D, N, r_t; vectors will be denoted using lowercase boldface letters, such as $\mathbf{x}$, $\mathbf{m}$; matrices will be denoted using uppercase boldface letters, such as $\mathbf{S}$, $\mathbf{A}$. Table 1.1 provides a list of the description of the symbols used in this thesis.

[5] The application to face synthesis is not discussed in this thesis.

Table 1.1: Notations

Notation	Description
N	Number of data points
D	Dimension of the original data space
$\mathbf{L}_i$	Set of data points belonging to Class i
$\mathbf{S}_t$, $\mathbf{S}_b$, $\mathbf{S}_w$	Total, between-class, within-class scatter matrices
$\tilde{\mathbf{S}}_t$, $\tilde{\mathbf{S}}_b$, $\tilde{\mathbf{S}}_w$	Whitened total, between-class, within-class scatter matrices
$\mathbf{H}_t$, $\mathbf{H}_b$, $\mathbf{H}_w$	Total, between-class, within-class precursor matrices
$\tilde{\mathbf{H}}_t$, $\tilde{\mathbf{H}}_b$, $\tilde{\mathbf{H}}_w$	Whitened total, between-class, within-class precursor matrices
r_t	Rank of original total scatter matrix
r_w	Rank of original within-class scatter matrix
r_b	Rank of original between-class scatter matrix
$\mathbf{a}$	Original data point
$\mathbf{m}$	Centroid of data
$\mathbf{m}_i$	Centroid of data points belonging to Class i
$\mathbf{1}$	Vector with all ones
$\mathbf{0}$	Vector with all zeros
$\mathbf{A}$	Matrix whose columns are data points
$\mathbf{U}$	Eigenvector matrix
$\mathbf{D}$	Eigenvalue matrix
$\mathbf{M}_p$	Labeling matrix whose columns denote the class labels of corresponding data points
$\mathbf{V}_1$	Eigenvector matrix for the *Identity Space*
$\mathbf{V}_3$	Eigenvector matrix for the *Variation Space*
$\mathbf{P}$	Final projection matrix of an algorithm

Chapter 2

Literature Survey

2.1 Face Recognition Algorithms

The past thirty years has been a prolific period for research on face recognition. A wide range of machine learning techniques have been proposed and experimented for this application. In this section, two categories of algorithms are reviewed: subspace-based (Section 2.1.1 to 2.1.5) and template-based algorithms (Section 2.1.6 and 2.1.7).

In the first category, face image are treated as vectors of pixel intensities which lie on a high-dimensional space, and statistical modeling is used to project these vectors to lower-dimensional subspaces which contain information for classification. Face recognition is then performed in these subspaces. In the second category, face images are characterized by a certain template and a set of parameters and face recognition is performed based on the parameters.

2.1.1 Eigenface

Turk and Pentland [27] proposes to use Principal Component Analysis (PCA) to reduce the dimension of face image vectors and to represent them using only a small number of principal components, called *Eigenfaces*. The technique of *PCA* suggests a compact representation of high-dimensional data by performing a dimensionality-reducing linear projection that maximizes the scatter of data samples. This feature is naturally appealing for appearance-based face recognition which compares 2D faces pixel-wise and treats them as high-dimensional vectors. It is also widely used as a preprocessing tool to perform dimensionality reduction prior to classification analysis.

The mathematical formulation of *PCA* is the following. Let column vectors $\{\mathbf{x}_i\}_{(i=1..N)}$ denote the image vectors from the gallery, which make up the columns of data matrix $\mathbf{X}$ i.e. $\mathbf{X} = [\mathbf{x}_1\mathbf{x}_2 \cdots \mathbf{x}_N]$. Let $\mathbf{m}$ denote the mean of $\{\mathbf{x}_i\}$, i.e. $\mathbf{m} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$. The scatter matrix of $\mathbf{X}$ is defined by

$$\mathbf{S_X} = \sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^\top \tag{2.1}$$

Then we seek a projection transformation matrix $\mathbf{W}$ such that the scatter of the transformed data ($\mathbf{y} = \mathbf{W}^\top(\mathbf{x} - \mathbf{m})$) is maximized. Thus we have the following criterion function:

$$\begin{aligned}\mathbf{W} &= \arg\max_{|\mathbf{W}|=1} |\mathbf{S_Y}| \\ &= \arg\max_{|\mathbf{w}|=1} |\mathbf{W}^\top\mathbf{S_X}\mathbf{W}| \end{aligned} \tag{2.2}$$

It can be shown that the solution of Equation 2.2 are in fact the matrix consisting of the eigenvectors of $\mathbf{S_X}$, which can be found by diagonalizing $\mathbf{S_X}$[1]:

$$\mathbf{S_X} = \mathbf{UDU}^\top$$

Here $\mathbf{U}$ is an orthogonal matrix consisting of the eigenvectors of $\mathbf{C}$, and $\mathbf{D}$ is a diagonal matrix consisting of corresponding eigenvalues. The eigenvalues are non-negative and sorted in decreasing order.

Thus the projection matrix $\mathbf{W}$ is:

$$\mathbf{W} = [\mathbf{u}_1\mathbf{u}_2 \cdots \mathbf{u}_k] \tag{2.3}$$

where $\mathbf{u}_j$ ($j = 1, \ldots, k$) are the first k columns of $\mathbf{U}$ which contain the most energy[2].

[1] $\mathbf{S_X}$ is a real symmetric matrix, so it is always diagonaliable

[2] The energy of an eigenvector is measured by the corresponding eigenvalue.

The low-dimensional representation of $\mathbf{X}$ is:

$$\mathbf{Y} = \mathbf{W}^\top(\mathbf{X} - \mathbf{M}) \tag{2.4}$$

where $\mathbf{M} = [\mathbf{m} \cdots \mathbf{m}]$ is the matrix whose columns consist of the mean of $\mathbf{X}$. And given a low-dimensional representation, the reconstruction (i.e. the mapping to the original data space) is the following formula:

$$\mathbf{X} = \mathbf{W}\mathbf{Y} + \mathbf{M} \tag{2.5}$$

Figure 2.1 shows examples of seven eigenfaces constructed from 15 face images of the publicly available Yale Face Database[3].

Two important properties of *PCA* need to be highlighted:

- **Minimum Squared Error.** PCA has the minimum mean squared reconstruction error among all linear transformations[4]. More precisely, for any linear transformation $\hat{\mathbf{W}}$, the mean squared reconstruction error is always greater than or equal to that of PCA, i.e.

$$E[||\epsilon_{\hat{\mathbf{W}}}||^2] \geq E[||\epsilon_{PCA}||^2] \tag{2.6}$$

- **Decorrelation.** Since the covariance matrix of $\mathbf{Y}$ is diagonal by construction, any two different elements $\mathbf{y}_i, \mathbf{y}_j (i \neq j)$ are uncorrelated. That means mutual correlation between any two samples in $\mathbf{X}$ is removed by the projection.

Although PCA provides the best low-dimensional approximation of the original data, it is not optimal for classification since it does not make use of any class information. However, Eigenface has inspired many researchers and acted as the basis of many methods. It is often used as a benchmark for comparison purpose.

[3] http://cvc.yale.edu/projects/yalefaces/yalefaces.html

[4] A linear transformation $\mathbf{P}$ may not necessarily be a projection, characterized by $\mathbf{P}^2 = \mathbf{P}$. We shall see such an example in later chapters: the whitening transform described in Section 3.2 is a linear transform which is not a projection.

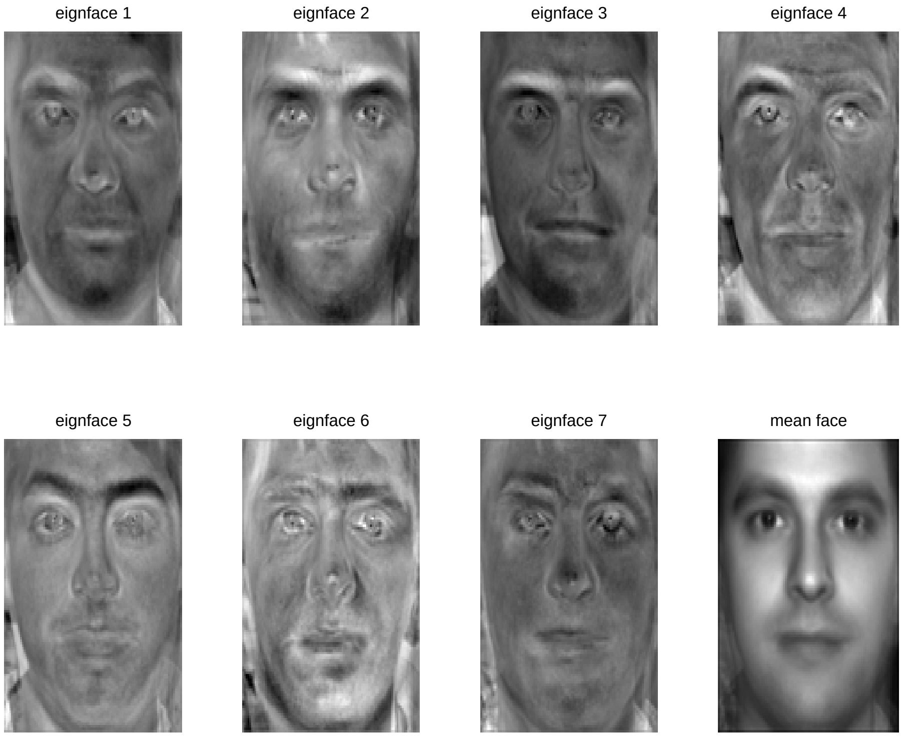


Figure 2.1: Seven eigenfaces constructed from 15 face images in Yale Face Database. They are ordered in decreasing order of their eigen-energy. The last image is the mean face.

2.1.2 Fisherface

Belhumeur et al. [2] propose to use Fisher Discriminant Analysis (*FDA*) to perform face recognition. *FDA* belongs to a more general type of methods, Linear Discriminant Analysis (*LDA*), which aims to find a projection direction to best separate the classes. More specifically, *FDA* uses the *Fisher Criterion* as the objective function which measures the degree of separation between classes .

The mathematical formulation is the following: each gallery image is denoted by a vector $\mathbf{x}_i$ ($i = 1, 2, \ldots, N$), each belonging to exact one class L_j ($j = 1, \ldots, C$). The number of data samples in Class L_j is denoted by N_j. Obviously, $N = \sum_{i=1}^{c} N_i$. The mean of class L_j is denoted by $\mathbf{m}_j$ i.e. $\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x}_i \in L_j} \mathbf{x}_i$. The mean of all data

$\mathbf{m} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$. The between-class scatter matrix $\mathbf{S}_b$ and the within-class scatter matrix $\mathbf{S}_w$ are defined by

$$\mathbf{S}_b = \sum_{i=1}^{c} n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T \tag{2.7}$$

$$\mathbf{S}_w = \sum_{i=1}^{c}\sum_{\mathbf{x}\in L_i}(\mathbf{x} - \mathbf{x}_i)(\mathbf{x} - \mathbf{m}_i)^T \tag{2.8}$$

The objective function of *FDA* is the following:

$$\mathbf{W} = \arg\max_{|\mathbf{W}|=1} \frac{|\mathbf{W}^\top \mathbf{S}_b \mathbf{W}|}{|\mathbf{W}^\top \mathbf{S}_w \mathbf{W}|}$$

Equation 2.1.2 can be interpreted as maximizing the between-class scatter of the projected samples (the numerator) and while minimizing the within-class scatter (the denominator).

It can be shown that under certain conditions[5], the column vectors of the solution to Equation 2.1.2 are in fact solutions to the following equation:

$$\mathbf{S}_w^{-1}\mathbf{S}_b\mathbf{w} = \lambda\mathbf{w} \tag{2.9}$$

i.e. $\mathbf{w}$ is an eigenvector of $\mathbf{S}_w^{-1}\mathbf{S}_b$, and λ is the corresponding eigenvalue.

In practice, there are two common shortcomings of *FDA*:

- The Small Size Size problem (SSS) [16]. When the number of training samples are small compared to the dimension, the scatter matrices (particularly $\mathbf{S}_w$) would be singular and Equation 2.9 would be ill defined.
- The Overfitting problem. Empirically *FDA* has shown to be very sensitive to the sampling power of the training data, and if the underlying distribution of the whole data space is very different from that of the training data, *FDA* would very probably perform badly.

[5] Particularly, the non-singularity of $\mathbf{S}_w$

Table 2.1: Comparison of PCA and LDA w.r.t common classification issues

Method	Objective Function	Representation Ability	Discrimination Ability	Over-fitting problem	Sensitivity to sample size
PCA	$\arg\max\limits_{\lvert\mathbf{W}\rvert=1} \lvert\mathbf{W}^\top \mathbf{S}_t \mathbf{W}\rvert$	Optimal	Sub-optimal	No	No
LDA	$\arg\max\limits_{\lvert\mathbf{W}\rvert=1} \frac{\lvert\mathbf{W}^\top \mathbf{S}_b \mathbf{W}\rvert}{\lvert\mathbf{W}^\top \mathbf{S}_w \mathbf{W}\rvert}$	N.A.	Optimal	Yes	Yes

Comparison: Eigenface vs Fisherface While *FDA* seeks a transformation matrix to maximize the ratio between the between-class scatter and within-class scatter, PCA seeks a transformation matrix maximizing the total scatter of projected data (which is the sum of between-class scatter and within-class scatter).

It is generally believed that for face recognition, algorithms based on LDA are superior to those based on PCA. However, [21] shows that PCA could outperform LDA when the training data set is small and that PCA is less sensitive to whether the training set samples well the underlying distribution.

Table 2.1 provides a summary of the relative strengths and weaknesses of PCA and LDA.

2.1.3 FFKT

Zhang and Sim [35, 36] proposes to perform *FDA* in the *whitened* data space, in which the *Fisher Criterion* can be easily evaluated. They also show that by working in the whitened data space, many previous discriminant subspace algorithms (such as *PCA + Null Space*) can be analyzed in a unifying framework.

As this technique is intimately related to the theory of this thesis, we shall the delay the detailed explanation to Chapter 4.

2.1.4 Tensorface

Vasilescu and Terzopoulos [29, 30, 31]propose to model face images within the framework of multi-linear algebra. Multi-linear algebra is a high-order generalization of linear algebra. Correspondingly, the basic element is the tensor, a generalization of vectors (1^{st} order tensor) and matrices (2^{nd} order tensor), in which points are indexed by N parameters. Vasilescu and Terzopoulos proposes to model the ensembles of face images by an $N-$mode tensor, in which each mode corresponds to a particular attribute of the face image (e.g. person identity, expression, etc). For example, a face image is uniquely determined, given the person's identity, expression, lighting conditions, etc. The Singular Value Decomposition (SVD) for matrices also admit a generalized version for tensors, called the $N - mode$ SVD, which orthogonalizes the N mode spaces[6] and decomposes the tensor as the $Mode - N$ product of N orthogonal spaces, as follows:

$$\mathcal{D} = \mathcal{Z} \times_1 \mathcal{U}_1 \times_2 \mathcal{U}_2 \cdots \times_N \mathcal{U}_N \tag{2.10}$$

This operation enables

- **Dimension reduction** The truncation of insignificant components of $\mathcal{U}_i$ gives a reduced model of the data set (although this is not optimal unlike the matrix-SVD case).

- **Independent manipulation of coefficients of different modes** This is because the $\mathcal{U}_i$ are mutually orthogonal after the $N - mode$ SVD. This property is particularly useful for applications such as face image synthesis and face expression transfer.

2.1.5 Kernel Methods

As an attempt to circumvent the limitation of linear methods in a non-linear data space, kernel methods (typically *KPCA* [22, 40] or *KLDA*[1, 20, 33] have been proposed to uncover the nonlinear structure embedded in the data space of face images. This is done by computing the higher order statistics, instead of relying on second-order statistics as linear methods do. More specifically, the data are mapped to a (usually high-dimensional) feature space and the inner product of points in the feature space is defined by a certain *kernel function* of points in the original data space. Once the inner product is defined, *PCA* or *LDA* can then be performed on the feature space without explicitly knowing the kernel functions. Common kernels include Gaussian, polynomial and Sigmoid functions.

[6] As we shall see in later chapters, the statistical interpretation of orthogonalization in linear algebra is *decorrelation*.

Table 2.2: Comparison of linear and kernel methods

Method	Information used	Sensitivity of performance to parameters/kernels	Reconstruction	Interpretability
Linear	First two moment statistics	No	Easy	Easy
Kernel	High-Order-Statistics (HOS)	Yes	Hard	Hard

Kernel methods can be seen as a generalization of existing linear modeling methods in that different kernels may be applied to represent different structure of the underlying space. *PCA* and *LDA* are special cases of kernel methods when the kernel is chosen to be the first-order polynomial. However, as the underlying structure of the space of face images is still unknown, the proper choice of the kernel function and corresponding parameters for face image modeling remains unknown and can only be tested and decided empirically. For applications such as face recognition, the performance of kernel methods are in general very sensitive to the choice of kernel functions and parameters. Huang et al. [15] discuss about optimal choices of the kernel parameters.

Unlike appearance-based modeling, template-based modeling methods characterize face images using a certain template model, and a set of parameters. Thus within each model, a face image is uniquely determined once given the set of parameters.

2.1.6 AAM

An active appearance model (AAM) [6, 7] is a statistical model which consists of a shape model and a gray-level appearance model. The AAM is constructed from a set of exemplar training images with labeled landmark points.

The shape model is computed by applying PCA on all the shape data (points coordinates), and the appearance model is built by applying another PCA on the sampled grey-level values of the interested region (area surrounded by the marked points). Then a united model is constructed by applying PCA to the concatenation of the shape model parameters and the appearance model parameters. A face image is thus uniquely determined given a the *AAM* parameters and new images can be

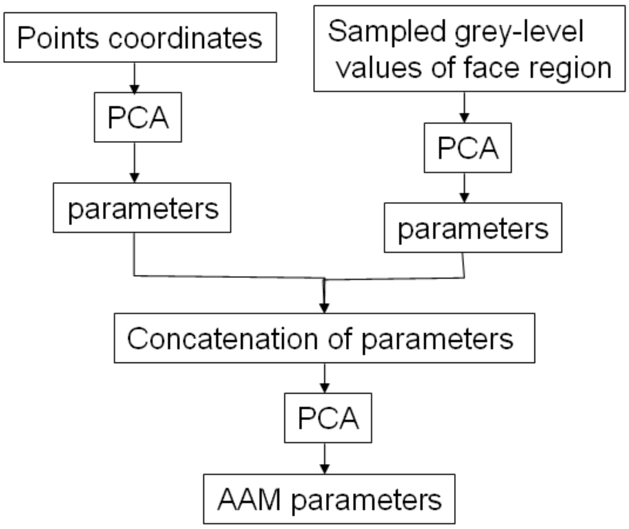

Points coordinates
Sampled grey-level values of face region
PCA
PCA
parameters
parameters
Concatenation of parameters
PCA
AAM parameters

Figure 2.2: Illustration of AAM construction. Figure extracted from [14].

synthesized by specifying the parameter values. Figure 2.2 shows the procedure of *AAM* model building.

To represent a given face image, an iterative search is used to obtain an optimal *AAM* parameter which minimizes the error between the given image and the one synthesized by AAM. From a certain starting point (usually the mean shape and mean appearance), the parameters are refined iteratively until convergence. This procedure assumes implicitly that the given image can be written as a linear combination of those in the database in terms of their shape coordinates and texture values. This requires a large training data set which is representative of face images that the model is likely to encounter. In [7], a training set of 400 images of faces is used to construct the AAM. Each image in the set is labeled with 122 points. The generated shape model is consisted of 23 parameters and appearance model 113 parameters. The final AAM has 80 parameters. Once a given face image is properly represented, face recognition could be performed on the *AAM* parameters using classification methods such as *LDA*.

Costen et al. ([8]) further attempts to study the facial variation subspaces within the space defined by *AAM*. They apply an iterative algorithm to find functional face subspaces which span sets of faces which vary in different ways.

Table 2.3: Comparison of subspace-based and template-based algorithms

Algorithms	Pros	Cons
Subspace-based	• Implementation is straightforward • Time and space complexities are relatively low • Size of training data set is scalable	• Performance deteriorates if training data set samples badly the underlying distribution • Performance deteriorates if assumptions regarding the underlying distribution fail • Sensitive to registration errors
Template-based	• Make use of domain knowledge • Face synthesis is easy	• (Much) Manual marking is needed • Training data set has to be sufficiently big • Time and space complexities are high

2.1.7 3D Morphable Model

The *3D morphable model* is the 3D counterpart of the 2D *AAM* and is created using a 3D face database of laser scans([3, 4, 5]). It consists of a shape model, built upon the 3D coordinates on the face surface, and a texture model, built upon the color values of every 3D point. Given a new face image, the model fitting process is a 2D-to-3D fitting, thus additional rendering parameters need to be included in the model. A particular advantage of the method is that it allows easier and more flexible modeling and manipulation of pose and illumination variations of faces compared with 2D models.

2.2 Dimension Reduction Algorithms

The problem of dimensionality reduction regularly arise in the fields of science and engineering, when there are large volumes of high-dimensional data. There are in general two objectives for performing dimension reduction:

- Reduce space storage
- Facilitate understanding and visualization of the (potentially complex) data structure

The two objectives are related in that they both involve a mapping from a high-dimensional (high-dim) space to a low-dimensional (low-dim) one. However, they differ on a very important ground. The emphasis of the former is on reconstruction but not representation: it requires that the original data vectors could be (faithfully) reconstructed from the low-dim mapping, and is not concerned with how well the low-dim mapping represent the original data structure. In contrast, the emphasis of the later is on representation rather than reconstruction: it seeks a low-dim representation to uncover the embedded structure hidden in the high-dimensional observations, but does not requires reconstruction. In other words, **reconstruction** (the first objective) requires an additional mapping from the low-dim space to the original high-dim one, whereas representation does not; **representation** (the second objective) requires that the internal structure of the original data distribution be maximally preserved in the low-dim space, whereas reconstruction does not.

Different algorithms have been proposed to cater for different objectives. Subspace-based modeling, a population technique for *reconstruction*-oriented dimension reduction, is discussed in Section 2.2.1. Manifold learning which is population for *representation*-oriented dimension reduction, is discussed in Section 2.2.2, 2.2.3 and 2.2.4.

2.2.1 PCA

PCA has been discussed is Section 2.1.1 as a face recognition technique. However, as also noted in Section 2.1.1, *PCA* is optimal for representation but not classification.

2.2.2 MDS

Multidimensional Scaling (*MDS*) [9] covers a variety of techniques in the area of multivariate data analysis. In manifold learning for face images, we are mainly interested in one of them: classical *MDS*, which is a metric-based technique which aims to uncover the embedded structure hidden in the high-dim observations based on interpoint distances. More precisely, it seeks a low-dim mapping so that Euclidean distances in the low-dim space best preserve the original interpoint distances.

Mathematically[9], given a distance matrix[10] $\mathbf{D_X}$ of the high-dim vectors $\mathbf{X}$, *MDS* seeks a low-dim data matrix $\mathbf{Y}$ which minimizes the cost function

$$\mathbf{Y} = \arg\min ||\tau(\mathbf{D_X}) - \tau(\mathbf{D_Y})||_{L^2} \tag{2.11}$$

where $\mathbf{D_Y}$ denotes the matrix of Euclidean distances of vectors in $\mathbf{Y}$ and the τ operator converts distances to inner products and is defined by $\tau(\mathbf{D}) = -\frac{\mathbf{HSH}}{2}$, with $\mathbf{S}$ being the matrix of squared distances and $\mathbf{H}$ is the "centering matrix".

It is proved in [34] that *MDS* is equivalent to *PCA* when the distance metric in the original high-dim space is Euclidean. In that case, the solution to Equation 2.11 is in fact the principal component scores of *PCA*.

2.2.3 Isomap

Isomap [26] builds upon classical *MDS* using the geodesic, or shortest-path, distance as the distance metric for the high-dim vectors. It consists of three steps:

- For each point, determine its neighborhood points on the manifold and represent the relations in a weighted graph,
- Estimate the geodesic distances between all pairs of points by computing their shortest path distances in the weighted graph,
- Apply classical *MDS* to matrix of geodesic distances to get a low-dim representation.

In sum *Isomap* seeks to capture the intrinsic geometry of the data by computing the geodesic manifold distances and then uses *MDS* to find a low-dim mapping to preserve this geometric structure. Since it is built upon geodesic manifold distances, *MDS* is capable of uncovering nonlinear structures hidden in the high-dim space.

[9] The mathematical formulation here is based upon that in [26].

[10] i.e. $\mathbf{D_X}(i, j) = ||\mathbf{x}_i - \mathbf{x}_j||_F$, where F denotes a certain distance metric, such as Euclidean distance.

Table 2.4: Comparison of subspace-based modeling and manifold learning

Approaches	Pros	Cons
Subspace-based Modeling	• Computation is efficient • Reconstruction is easy • Uncover statistical relationships among data points	• Assumptions about the underlying structure of data may not be true
Manifold learning	• Uncover the embedded nonlinear structure	• Cannot reconstruct samples based on low-dim mapping • Need sufficient samples to correctly model the structure

2.2.4 LLE

Locally Linear Embedding (*LLE*) finds a low-dim mapping that best preserves the local neighborhood of each point. More specifically, *LLE* consists of three steps:

- For each point, determine its neighborhood,
- Compute the weights that best reconstruct each point as a linear combination of its neighboring points,
- Compute the low-dim embedding vectors best reconstructed by the weights.

In sum *Isomap* seeks to capture the intrinsic geometry of the data using local neighborhoods and these overlapping local neighborhoods collectively provide information for the global geometry. Since it is built upon local neighborhoods, *MDS* is also capable of uncovering nonlinear structures hidden in the high-dim space.

2.2.5 Discussion

Table 2.4 summarizes the strengths and weaknesses of the two categories of methods discussed in this section.

2.3 Summary

Table 2.5 summarizes the algorithms reviewed in this section in terms of their applications. For comparison purposes, we have also listed our method *MMDA* in the table.

We have used three suitability levels to describe the application of different algorithms to different tasks: *Good*, *Acceptable*, *Poor*. By *Good*, we mean that either the method has been specifically designed for a certain application, or the method has shown empirically satisfactory performance. By *Acceptable*, we mean that the method was not designed for, but can be applied to a certain task, and the performance is acceptable but not optimal. By *Poor*, we mean that the method has not been designed for the task and is rarely used for that particular task.

For the task of classification, appearance-based methods *LDA* and *FFKT* and template-based methods *AAM* and *3D MM* have generally shown reasonably good performance. However, we note that it would not be fair to compare the performance of appearance-based and template-based methods since they different inputs and have different time complexities. *PCA* was initially designed for data representation and not classification, but it is now wildly used in the face recognition literature and has become a benchmark in the field. There has been extensive research of the application of *Kernel PCA* and *Kernel LDA* to face recognition, but the performance is largely dependent on the choice of kernel functions and parameters. Both *Isomap* and *LLE* have been tested for face recognition, but these attempts have mostly been exploratory.

For the task of dimension reduction, *PCA* has been proved to provide the best lower-dimensional representation in terms of RMSE[11] among all linear transformations and it is natural to do reconstruction with *PCA* since it is simply an orthogonal projection. Manifold methods *MDS*, *Isomap* and *LLE* have mainly been designed to visualize the structure of the data (i.e. for representation), but reconstruction is not straight-forward since the mapping is nonlinear. *Tensorface* and *MMDA* have shown reasonably good performance for image representation and reconstruction is straight-forward since they use multi-linear or linear representation. Finally, *LDA*, *FFKT* and *Kernel PCA/LDA* are rarely used for dimension reduction.

[11] I.e., root mean square error

We have used two levels to denote the relative complexities of different methods. *Tensorface*, *AAM* and *3D MM* require optimization procedures and are much more time-consuming compared to other methods, of which the implementation is rather straight-forward.

By comparison, we see that our method *MMDA* allows a good compromise between different tasks and time complexity.

Table 2.5: Summary of algorithms and their suitability for different tasks

Algorithms			**Suitability for Classification**	**Suitability for Dimension Reduction**		**Time Complexity**
				Representation	Reconstruction	
Appearance-based	Subspace	PCA	Acceptable	Poor	Poor	Low
		LDA	Good	Poor	Poor	Low
		FFKT	Good	Poor	Poor	Low
		KPCA/KLDA	Acceptable	Poor	Poor	Low
		TensorFace	Acceptable	Acceptable	Acceptable	High
		MMDA	Good	Acceptable	Acceptable	Low
	Manifold	MDS	Poor	Good	Poor	Low
		Isomap	Poor	Good	Poor	Low
		LLE	Poor	Good	Poor	Low
Template-based	AAM		Acceptable	Poor	Poor	High
	3D MM		Acceptable	Poor	Poor	High

Chapter 3

Basic Concepts

3.1 Scatter matrices and Precursor matrices

3.1.1 Definition

Let $\mathbf{A} = \{a_1, \ldots, a_N\}$, $\mathbf{a_i} \in \mathbb{R}^D$ denote a data set of given D-dimensional vectors. Each vector belongs to exactly one of the C classes $\{L_1, \ldots, L_C\}$. The number of vectors in class L_i is denoted by N_i, thus we have $N = \sum_{i=1}^{C} N_i$. The between-class scatter matrix $\mathbf{S}_b \in \mathbb{R}^{D \times D}$, the within class scatter matrix $\mathbf{S}_w \in \mathbb{R}^{D \times D}$ and the total scatter matrix $\mathbf{S}_t \in \mathbb{R}^{D \times D}$ are defined as follows:

$$\mathbf{S}_t = \sum_{i=1}^{N} (\mathbf{a}_i - \mathbf{m})(\mathbf{a}_i - \mathbf{m})^\top = \mathbf{H}_t \mathbf{H}_t^\top \tag{3.1}$$

$$\mathbf{S}_b = \sum_{i=1}^{C} N_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^\top = \mathbf{H}_b \mathbf{H}_b^\top \tag{3.2}$$

$$\mathbf{S}_w = \sum_{i=1}^{C} \mathbf{S_w^i} = \sum_{i=1}^{C} \sum_{\mathbf{a} \in L_i} (\mathbf{a}_i - \mathbf{m}_i)(\mathbf{a}_i - \mathbf{m}_i)^\top = \mathbf{H}_w \mathbf{H}_w^\top \tag{3.3}$$

where $\mathbf{S_w^i}$ denotes individual within-class scatter matrices, $\mathbf{m}_i$ denotes the mean of vectors in Class i and $\mathbf{m}$ is the global mean of A. The matrices $\mathbf{H_b} \in \mathbb{R}^{D \times C}$ and $\mathbf{H_w} \in \mathbb{R}^{D \times N}$, and $\mathbf{H_t} \in \mathbb{R}^{D \times N}$ are the *precursor* matrices of the between-class scatter matrix, the within-class scatter matrix and the total scatter matrix respectively,

$$\mathbf{H}_b = [\sqrt{N_1}(\mathbf{m}_1 - \mathbf{m}), \ldots, \sqrt{N_C}(\mathbf{m}_C - \mathbf{m})], \tag{3.4}$$

$$\mathbf{H}_w = [\mathbf{A}_1 - \mathbf{m}_1 \cdot \mathbf{1}_1^\top, \ldots, \mathbf{A}_C - \mathbf{m}_C \cdot \mathbf{1}_C^\top], \tag{3.5}$$

$$\mathbf{H}_t = [\mathbf{a}_1 - \mathbf{m}, \ldots, \mathbf{a}_N - \mathbf{m}]. \tag{3.6}$$

Here, $\mathbf{1}_i = (1, \ldots, 1)^\top \in \mathbb{R}^{N_i}$ and A_i is the data matrix for class L_i.

It can be easily proved that

$$\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w \tag{3.7}$$

3.1.2 Rank

The *rank* of a matrix **A** is defined as the maximal number of linearly independent columns of **A**. This is equivalent to say that the rank of a matrix is the dimension of the space spanned by its column vectors. As we shall see throughout the thesis, this quantity is of particular interest to us since it conveys information about the dimensionality of a given set of data vectors.

For high-dimensional data (i.e. $D > N$), we have the following observations with respect to the rank of scatter and precursor matrices[1]:

$$\mathrm{Rank}(\mathbf{S_b}) = \mathrm{Rank}(\mathbf{H_b}) \leq C - 1 \tag{3.8}$$

$$\mathrm{Rank}(\mathbf{S_w}) = \mathrm{Rank}(\mathbf{H_w}) \leq N - C \tag{3.9}$$

$$\mathrm{Rank}(\mathbf{S_t}) = \mathrm{Rank}(\mathbf{H_t}) \leq N - 1 \tag{3.10}$$

The proof is provided in Theorem A.4 in Appendix A.

Remark 3.1.1. *For all the scatter and corresponding precursor matrices defined in 3.1.1, we have Range*$(\mathbf{S})$ = *Range*$(\mathbf{H})$ [2].

[1] For detailed proof, please refer to Appendix A.

[2] The range of matrix is defined as the space spanned by it columns.

Using Equations 3.8 and 3.1, the proof is straight-forward and is omitted here. We shall use the following notations throughout the thesis:

$$r_b := \mathrm{Rank}(\mathbf{S}_b) \tag{3.11}$$

$$r_w := \mathrm{Rank}(\mathbf{S}_w) \tag{3.12}$$

$$r_t := \mathrm{Rank}(\mathbf{S}_t) \tag{3.13}$$

with

$$r_b \leq C - 1 \tag{3.14}$$

$$r_w \leq N - C \tag{3.15}$$

$$r_t \leq N - 1 \tag{3.16}$$

3.1.3 Efficient Eigen-Decomposition

Scatter matrices defined in 3.1.1 are all positive semi-definite, thus they always yield an *eigen-decomposition* [3]. More precisely, all scatter matrices can be written in the form of

$$\mathbf{S} = \mathbf{H}\mathbf{H}^\top = \begin{bmatrix} \mathbf{U} & \mathbf{U}_\perp \end{bmatrix} \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{U}^\top \\ \mathbf{U}_\perp^\top \end{bmatrix} = \mathbf{U}\mathbf{D}\mathbf{U}^\top \tag{3.17}$$

where $\mathbf{D}$ is a diagonal matrix containing the positive[4] *eigenvalues* of $\mathbf{S}$, and $\mathbf{U}$ and $\mathbf{U}_\perp$ are orthogonal matrices[5] containing the eigenvectors of $\mathbf{S}$.

For high dimensional data (i.e.,$D \gg N$), the size $S \in \mathbb{R}^{D\times D}$ is huge, and the computation of its eigen-decomposition is extremely resource-consuming. On the other hand, a related matrix defined as

$$\bar{\mathbf{S}} = \mathbf{H}^\top\mathbf{H} \tag{3.18}$$

[3] which is closely related to the *singular value decomposition* of precursor matrices.

[4] All eigenvalues of **S** are non-negative because it is positive semi-definite.

[5] In thesis, we shall call a matrix *orthogonal* if its column vectors are orthogonal with each other, although the mathematical definition for *orthogonal matrix* is much stricter.

has a much smaller size and remains positive semi-definite. If there is a simple link between the eigenvectors and eigenvalues of $\mathbf{S}$ and $\mathbf{R}$, then $\mathbf{S}$ can be eigen-decomposed much more efficiently. Remark 3.1.1 establishes this link.

Remark 3.1.1. $\mathbf{x}$ *is an eigenvector of the matrix* $\mathbf{S} = \mathbf{H}\mathbf{H}^\top$ *corresponding to a positive eigenvalue* $\lambda > 0$ *if and only if* $\mathbf{H}^\top\mathbf{x}$ *is an eigenvector of the matrix* $\bar{\mathbf{S}} = \mathbf{H}^\top\mathbf{H}$ *corresponding to the same eigenvalue* λ*. If* $\mathbf{x}$ *is a unit vector, then* $\mathbf{H}^\top\mathbf{x}$ *is of norm* $\sqrt{\lambda}$*.*

Based on Remark 3.1.1, we could choose to diagonalize $\mathbf{H}\mathbf{H}^\top$ or $\mathbf{H}^\top\mathbf{H}$, whichever is smaller. This would guarantee to find an orthogonal basis for Range($\mathbf{S}_t$), the space we are mainly interested in.

3.2 Whitening Transform

3.2.1 Definition

If the total scatter matrix S_t is eigen-decomposed in the following manner:

$$\mathbf{S_t} = \begin{bmatrix} \mathbf{U} & \mathbf{U}_\perp \end{bmatrix} \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} U^\top \\ U_\perp^\top \end{bmatrix} \tag{3.19}$$

where $\mathbf{D}$ is a diagonal matrix with positive decreasing eigenvalues of $\mathbf{S_t}$, and columns of $\mathbf{U}$ and $\mathbf{U}_\perp$ represent eigenvectors corresponding to positive and zero eigenvalues, then the whitening transform is defined as:

$$\mathbf{P} := \mathbf{U}\mathbf{D}^{-\frac{1}{2}} \tag{3.20}$$

The whitening transform is also called the **Fukunaga-Koontz Transform (FKT)** [12] [17] in the context of binary classification problems. We shall use the term *FKT* to denote the transform of Equation 3.20 when dealing with classification problems in later chapters.

3.2.2 Interpretations

The whitening transform is composed of two procedures:

- $\mathbf{U}$ rotates the coordinate system to decorrelate the axes, i.e. the new total scatter matrix $\tilde{\mathbf{S}}_t = \mathbf{P}^\top \tilde{\mathbf{S}}_t \mathbf{S}$ is a diagonal matrix. This procedure is exactly the same as in *PCA*.

- $\mathbf{D}^{-\frac{1}{2}}$ normalizes the axes so that each dimension has the same extent (or weight) in the new coordinate system, i.e. the diagonal entries of the new total scatter matrix are all equal $\tilde{\mathbf{S}}_t = \mathbf{P}^\top \mathbf{S}_t \mathbf{P} = \mathbf{I}$.

Similar to *PCA*, the whitening transform only decorrelates the axes (based on second-order statistics), which means that the new axes are not necessarily independent (with additional constraints on higher-order statistics) with each other. However, it is proved that [10] if the sample vectors follow a normal distribution, then the resulting axes are indeed independent with each other.

3.2.3 Algorithm

Algorithm 1 describes the whitening transform. The most expensive operations involved are matrix multiplication and eigen-decomposition. Thus the time complexity is $O(DN^2)$ or $O(ND^2)$ (whichever is smaller) and the space complexity is $O(DN)$.

Algorithm 1 Whitening transform of a given data matrix

```
Input: Data matrix $\mathbf{A} \in \mathbb{R}^{D \times N}$
Output: Whitened data matrix $\tilde{\mathbf{A}} \in \mathbb{R}^{r_t \times N}$.
  Compute $\mathbf{H}_t$ of data matrix $\mathbf{A}$.
  Compute the eigenvector matrix $\mathbf{U}$ and eigenvalue matrix $\mathbf{D}$ for $\mathbf{S}_t$ based on
    Remark 3.1.1 and compute the whitening transform $\mathbf{P} = \mathbf{U}\mathbf{D}^{-\frac{1}{2}}$.
  $\tilde{\mathbf{A}} = (\mathbf{P})^\top \mathbf{A}$.
```

3.2.4 Reverse whitening

For Chapter 4 and 5, we whiten the given data matrix before carrying out the analysis. Furthermore, we have to be able to transform the data from the whitened data space back to the original space. This is easily done using the following transform that we shall term as **Reverse whitening**:

$$\mathbf{Q} := \mathbf{U}\mathbf{D}^{\frac{1}{2}} \tag{3.21}$$

More precisely, a typical analysis framework in this thesis would be the following. Given a data matrix $\mathbf{A}$, we shall perform the following three procedures:

- Whiten the data using $\tilde{\mathbf{A}} = (\mathbf{P})^\top \mathbf{A}$, where $\mathbf{P}$ is defined in Equation 3.20 on Page 27.
- Perform analysis on $\tilde{\mathbf{A}}$ to obtain $\hat{\tilde{\mathbf{A}}}$.
- Reverse-whiten the data using $\hat{\mathbf{A}} = \mathbf{Q}\hat{\tilde{\mathbf{A}}}$.

$\hat{\mathbf{A}}$ would be the final output of the analysis.

3.2.5 Decomposition and Reconstruction

The rest of the thesis is mainly concerned with the decomposition and reconstruction in the whitened data space. We shall demonstrate in this section that there is a close relationship between the decomposition and reconstruction in the whitened and original space.

Since the whitening transform is a full-ranked linear transform, there is a one-to-one correspondence between the original data set and the whitened points. Thus any decomposition and reconstruction in the whitened space has a corresponding effect in the original space. A legitimate question to ask is whether the decomposition and reconstruction in the whitened space can be carried forward to the original space. In order to study this question, let's firstly investigate the relationship between the projections in the two spaces.

Figure 3.1 shows the effect of whitening and reverse-whitening procedures. Figure 3.1(a) shows data points generated from a Gaussian distribution of zero mean and a covariance matrix of [2,1;1,1]. Figure 3.1(b) shows whitened data points after the whitening procedure. Figure 3.1(c) shows projections of the whitened points onto two orthogonal directions. Figure 3.1(d) shows the effect of reverse-whitening the projected points back to the original space.

Thus we have the following important remark:

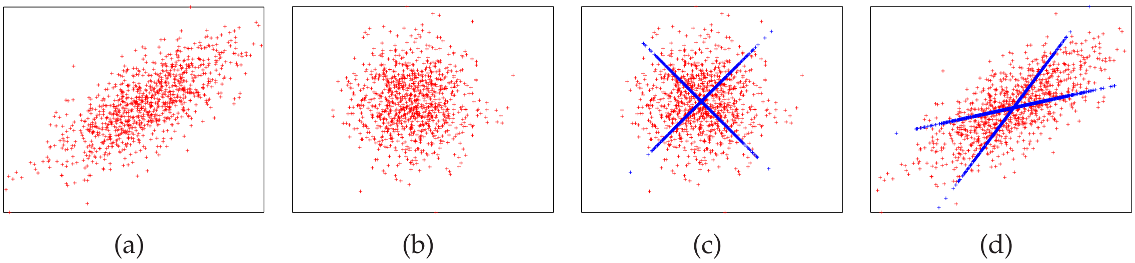


Figure 3.1: illustration of the whitening and reverse-whitening procedures. (a) Original data points following a Gaussian distribution. (b) Whitened data points (c) Projections of whitened data points to two orthogonal directions. (d) Projections reverse-whitened to the original data space.

Remark 3.2.1. ***Correspondence of Projections*** *In general, an orthogonal projection in the whitened data space corresponds to, in fact, an oblique projection in the original data space.*

Here, as well as in the rest of the thesis, we refer to *projection* in the strict linear-algebra sense: a linear mapping $\mathbf{P}$ is a projection if and only if we have $\mathbf{P}^2 = \mathbf{P}$. Intuitively, this means that multiple actions of the operation result in the same effect as one operation. A projection is orthogonal if and only if $\mathbf{P} = \mathbf{P}^\top$, otherwise it is an oblique projection.

Mathematically, we could derive the projection formula as follows. If we denote the whitening transform as $\mathbf{W}$, the orthogonal projection in the whitened space as $\mathbf{V}$, the reverse-whitening transform as $\mathbf{W}_r$, then the linear mapping is

$$\begin{aligned}\mathbf{P} &= \mathbf{W}_r\mathbf{V}\mathbf{V}^\top\mathbf{W}^\top && (3.22)\\ &= (\mathbf{U}\mathbf{D}^{\frac{1}{2}})\mathbf{V}\mathbf{V}^\top(\mathbf{U}\mathbf{D}^{-\frac{1}{2}})^\top && (3.23)\end{aligned}$$

To verify whether $\mathbf{P}$ is a projection, we perform the following computations:

$$\begin{aligned}\mathbf{P}^2 &= (\mathbf{U}\mathbf{D}^{\frac{1}{2}})\mathbf{V}\mathbf{V}^\top(\mathbf{U}\mathbf{D}^{-\frac{1}{2}})^\top(\mathbf{U}\mathbf{D}^{\frac{1}{2}})\mathbf{V}\mathbf{V}^\top(\mathbf{U}\mathbf{D}^{-\frac{1}{2}})^\top && (3.24)\\ &= (\mathbf{U}\mathbf{D}^{\frac{1}{2}})\mathbf{V}\mathbf{V}^\top\mathbf{D}^{-\frac{1}{2}}\mathbf{U}^\top\mathbf{U}\mathbf{D}^{\frac{1}{2}}\mathbf{V}\mathbf{V}^\top(\mathbf{U}\mathbf{D}^{-\frac{1}{2}})^\top && (3.25)\\ &= (\mathbf{U}\mathbf{D}^{\frac{1}{2}})\mathbf{V}\mathbf{V}^\top(\mathbf{U}\mathbf{D}^{-\frac{1}{2}})^\top && (3.26)\\ &= \mathbf{P} && (3.27)\end{aligned}$$

Here we have used the fact that $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$, since both matrices are orthogonal.

Therefore $\mathbf{P}$ is indeed a projection. Moreover, we could easily see that it is not an orthogonal projection, since $\mathbf{P}^\top \neq \mathbf{P}$.

Remark 3.2.1 implies that the decomposition and reconstruction are justified in the original space. In fact, if we carry out the following three steps:

- Apply whitening transform to the dataset,
- Perform component extraction and dimension reduction in the whitened space,
- Apply reverse whitening to project the points in the whitened space back to the original space,

Then the component extraction and dimension reduction in the whitened space correspond to the same operations in the original space, but through *oblique* projections (unlike *orthogonal* projections as in the whitened space).

Therefore, compared with *Isomap* or *LLE* which provide a good representation of the nonlinear data structure but which do not allow easy reconstruction based on the lower-dimensional representation, the method presented in this thesis could naturally handle reconstruction.

Chapter 4

Theoretical Foundation: Discriminant Subspace Analysis

The content of this chapter is built upon [36, 37, 38].

A common approach to classification problems in a high-dimensional data space is dimension reduction. By projecting the data to a lower-dimensional subspace, we wish that the class-discriminating information is preserved or even accentuated. The judicious choice of the subspace is thus critical. *FLD* is a traditional method commonly used for choosing an optimal subspace.

This chapter provides an analysis of *FLD* in a non-traditional context: the whitened data space. We shall see that insights can be obtained by examining the properties of scatter matrices in this space and we have a neat decomposition of the whitened space into subspaces with different discriminating power.

At the end of the chapter, we shall have a clear picture of what the whitened space looks like: its decomposition into orthogonal subspaces, each containing different degrees of discriminating information, and the structure of these subspaces.

In this chapter, we shall assume that the given data matrix has been whitened before any analysis is carried out.

4.1 Measure of Discriminating Power

In this section, we shall describe the criterion we use to measure the discriminating power of a subspace. We are going to see that the *Fisher* criterion can be written explicitly in the whitened data space in terms of eigenvalue ratios; thus we propose to use these eigenvalue ratios as the measure of discriminating power of any given subspace.

Firstly, Let's apply *FKT* (defined in 3.2) to whiten the data matrix **A**, we have

$$\begin{aligned} \tilde{\mathbf{A}} &= \mathbf{P}^\top \mathbf{A} && (4.1) \\ \tilde{\mathbf{S}}_b + \tilde{\mathbf{S}}_w &= \mathbf{P}^\top \mathbf{S}_b \mathbf{P} + \mathbf{P}^\top \mathbf{S}_w \mathbf{P} && (4.2) \\ &= \mathbf{P}^\top \mathbf{S}_t \mathbf{P} \\ &= \tilde{\mathbf{S}}_t \\ &= \mathbf{I} \end{aligned}$$

Remark 4.1.1. *We note that the rank of the scatter matrices are preserved after the whitening transform:*

$$\begin{aligned} Rank(\tilde{\mathbf{S}}_t \in \mathbb{R}^{r_t \times r_t}) &= r_t && (4.3) \\ Rank(\tilde{\mathbf{S}}_b \in \mathbb{R}^{r_t \times r_t}) &= r_b && (4.4) \\ Rank(\tilde{\mathbf{S}}_w \in \mathbb{R}^{r_t \times r_t}) &= r_w && (4.5) \\ & && (4.6) \end{aligned}$$

The proof is provided in Theorem A.4 in Appendix A.

If **x** is any eigenvector of $\tilde{\mathbf{S_b}}$, then

$$\begin{aligned} \tilde{\mathbf{S}}_b x &= \lambda x && (4.7) \\ \Rightarrow \tilde{\mathbf{S}}_w x &= (1 - \lambda) x && (4.8) \end{aligned}$$

This implies that $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$ share the same eigen-basis and their corresponding eigenvalues are complementary (i.e. their sum is equal to one). More precisely, we have the following eigen-decompositions:

Table 4.1: Direct sum decomposition of whitened data space

Subspaces with Eigenvalue ratio	**Basis vectors**	**Relationship to $\tilde{\mathbf{S}}_w$ and $\tilde{\mathbf{S}}_b$**	**Rank**
I ($\frac{\lambda_b^i}{\lambda_w^i} = \infty$)	$\{\mathbf{v}_i \mid \lambda_w^i = 0, \lambda_b^i = 1\}$	$\text{Null}(\tilde{\mathbf{S}}_w) \cap \text{Range}(\tilde{\mathbf{S}}_b)$	$r_t - r_w \leq C - 1$
II ($0 < \frac{\lambda_b^i}{\lambda_w^i} < \infty$)	$\{\mathbf{v}_i \mid 0 < \lambda_w^i, \lambda_b^i < 0\}$	$\text{Range}(\tilde{\mathbf{S}}_w) \cap \text{Range}(\tilde{\mathbf{S}}_b)$	$r_b + r_w - r_t \leq min(C-1, N-C)$
III ($\frac{\lambda_b^i}{\lambda_w^i} = 0$)	$\{\mathbf{v}_i \mid \lambda_w^i = 1, \lambda_b^i = 0\}$	$\text{Range}(\tilde{\mathbf{S}}_w) \cap \text{Null}(\tilde{\mathbf{S}}_b)$	$r_t - r_b \leq N - C$

$$\tilde{\mathbf{S}}_b = \mathbf{V}\Lambda_b\mathbf{V}^\top \tag{4.9}$$

$$\tilde{\mathbf{S}}_w = \mathbf{V}\Lambda_w\mathbf{V}^\top \tag{4.10}$$

$$\mathbf{I} = \Lambda_b + \Lambda_w \tag{4.11}$$

where $\mathbf{V} \in \mathbb{R}^{r_t \times r_t}$ is the orthogonal eigenvector matrix (i.e. whose columns $\{\mathbf{v_i}\}_{i=1,\ldots,r_t}$ are eigenvectors of $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$, $\mathbf{\Lambda}_b, \mathbf{\Lambda}_w \in \mathbb{R}^{r_t \times r_t}$ are diagonal eigenvalue matrices with non-negative entries[1].

If we denote the eigenvalues of $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$ corresponding to the eigenvector $\mathbf{v_i}$ as λ_b^i and λ_w^i (with $\lambda_b^i + \lambda_w^i = 1, \forall i = 1, \ldots, r_t$) , then the eigenvalue ratio $\frac{\lambda_b^i}{\lambda_w^i}$ can be shown to be equivalent to the *FLD* criterion in the whitened space [36]. This is our measure of discriminating power of the eigenvector $\mathbf{v}_i$.

4.2 Discriminant Subspaces

If we denote the eigenvalues of $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$ corresponding to the eigenvector $\mathbf{v_i}$ as λ_b^i and λ_w^i (with $\lambda_b^i + \lambda_w^i = 1, \forall i = 1, \ldots, r_t$) , we can partition the common eigenbasis $\{\mathbf{v_i}\}_{i=1,\ldots,r_t}$ according to the value ratio $\frac{\lambda_b^i}{\lambda_w^i}$. This is equivalent to decomposing the whitened data space in the form of direct sum [2] of subspaces *I, II and III*, as shown in Table 4.1.

[1] This is because covariance matrices are always semi-positive definite.

[2] If a space S is written as the direct sum of subspaces $S_1, \ldots, S_n$, then we have $S = S_1 + \cdots + S_2$, and $S_i \bigcap S_j = \emptyset$ for $i \neq j$. The direct sum is orthogonal if $S_i \perp S_j$ for $i \neq j$.

Theses subspaces are orthogonal to each other because their bases are orthogonal to each other.

4.3 Identity Space and Variation Space

4.3.1 Definition

Let's first coin two terms for *Subspaces I and III*, the rationale of which will become clear as we proceed to explore their properties.

$$\textbf{Identity Space} := \textit{Subspace I} \tag{4.12}$$

$$\textbf{Variation Space} := \textit{Subspace III} \tag{4.13}$$

Let's denote $\mathbf{V}_1$ and $\mathbf{V}_3$ as the matrices whose columns consist of the bases of the *Identity Space* and *Variation Space*.

4.3.2 Properties

Theorem 4.3.1 and 4.3.2 lay the foundation for our discriminant analysis of later chapters.

Theorem 4.3.1. ***Property of Identity Space*** *We have*

$$\mathbf{V}_1^\top \tilde{\mathbf{a}}_i = \mathbf{V}_1^\top \tilde{\mathbf{m}}_k, \forall \tilde{\mathbf{a}}_i \in L_k \tag{4.14}$$

or equivalently,

$$\mathbf{V}_1^\top (\tilde{\mathbf{a}}_i - \tilde{\mathbf{m}}_k) = 0, \forall \tilde{\mathbf{a}}_i \in L_k \tag{4.15}$$

The proof is detailed in [37].

From Equation 4.14, we see that in the *Identity Space*, all samples of Class L_k project onto the class mean $\mathbf{m}'_k = \mathbf{V}_1^\top \tilde{\mathbf{m}}_k \in \mathbb{R}^{r_t - r_w}$, and all within-class variations $(\tilde{\mathbf{a}}_i - \tilde{\mathbf{m}}_k)$[3] of Class L_k is projected to the zero vector. This means that the the quantity $\mathbf{m}'_k$ alone can be used represent the identity of Class L_k in the *Identity Space*. In other

[3] We note that they are the columns of $\tilde{\mathbf{H}}_w$

words, the *Identity Space* contains only between-class variation information, but no within-class variation information.

We shall coin the term:

$$\textbf{Identity Vectors} := \{\mathbf{m}'_k\}_{k=1,\dots,C} \tag{4.16}$$

to denote projection of the class samples onto the *Identity Space*. We know from 4.3.1 that all samples project to the *Identity vector* and thus in the *Identity Space*, each *Identity Vector* represents a class.

Theorem 4.3.2. ***Property of Variation Space*** *We have*

$$\forall\, k, \qquad \mathbf{V}_3^\top \widetilde{\mathbf{m}}_k = \mathbf{0}. \tag{4.17}$$

The proof is detailed in [38].

Theorem 4.3.2 states that within the *Variation Space*, the means of all classes are the same, i.e., the origin. Thus this subspace contains no information regarding the between-class variation. It only tells about how the vectors are distributed around the means, i.e. the within-class variations.

4.3.3 Geometric Structure

Due to the special properties of the *Identity Space* and *Variation Space*, we study the geometric structure of these two subspaces in this section.

Theorem 4.3.3. ***Geometric Structure of Identity Space*** *If there are equal number of samples in each class, then the identity vectors* $\mathbf{m_k}'$ *lie on a simplex, i.e., the magnitude of all identity vectors and the angles between them are the same.*

The proof is detailed in [37].

Figure 4.1 illustrates the geometric structure of the Identity Space.

Theorem 4.3.4. ***Geometric Structure of Variation Space*** *The Variation Space has the following structure:*

- *Projections of different classes to this subspace are orthogonal to each other.*

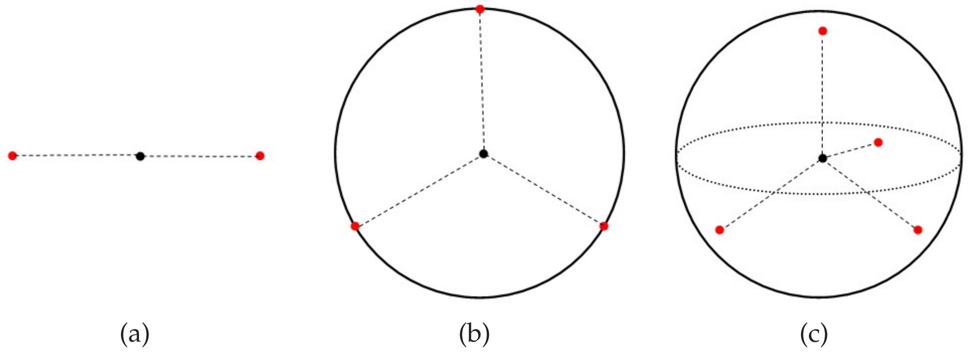


Figure 4.1: Illustration of the Identity Sphere. (a) When $C = 2$, it degenerates into a straight line and the identity vectors are ends of the line; (b) When $C = 3$, it is a circle in 2D space and the identity vectors are vertices of a regular triangle; (c) When $C = 4$, it is a sphere in 3D space and the identity vectors are vertices of a regular tetrahedron. Figure extracted from [37].

- *Projections of sample vectors from the same class to this subspace lie on a simplex, i.e. the magnitude of these vectors and angles between them are the same.*

The proof is detailed in [38].

Figure 4.2 shows an example of Variation Space for two classes $N_1 = 2$ and $N_2 = 3$.

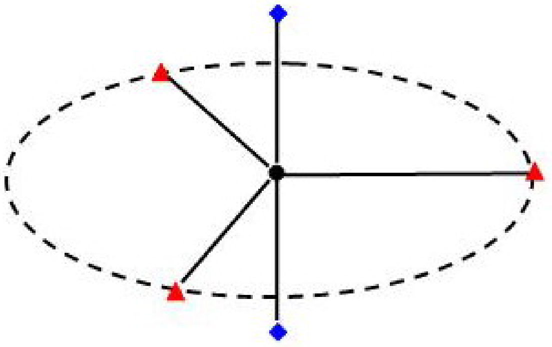


Figure 4.2: Illustration of the geometric structure of Variation Space. Note that different shapes (or colors) represent distinct classes. Class One has two data samples, and both are orthogonal to Class Two, which has three samples evenly distributed on a circle. Figure extracted from [38].

4.3.4 Conditions for Maximum Rank

We know from Table 4.1 on Page 34 that the maximum rank of the *Identity Space* and the *Variation Space* is $C - 1$ and $N - C$ respectively. As we have shown that the *Identity Space* is the most discriminating as it contains purely between-class information, we wish this subspace to exist to its maximum extent. Theorem 4.3.5 provides sufficient conditions for this situation.

Theorem 4.3.5. ***Conditions for Maximum Rank of Identity and Variation Space*** *The Identity Space and the Variation Space achieve their maximum rank if the following two conditions are satisfied:*

- $D \leq N - 1$
- *Null*$(\tilde{\mathbf{S}}_t) = \{\mathbf{1} \in \mathbb{R}^{r_t}\}$*, i.e. the sample vectors are linearly independent, except for the constraint that they sum up to the zero vector*[4]*.*

In this case, Subspace II doesn't exist and the whitened data space is composed of only Identity and Variation Space.

The proof is detailed in [37].

4.3.5 Decomposition of Space

From Table 4.1 we know that

$$\text{Subspace II} = \text{Range}(\tilde{\mathbf{S}}_b) \cap \text{Range}(\tilde{\mathbf{S}}_w) \tag{4.18}$$

If the conditions of Theorem 4.3.5 are satisfied, then *Subspace II* doesn't exist, which implies that

$$\text{Range}(\tilde{\mathbf{S}}_b) \cap \text{Range}(\tilde{\mathbf{S}}_w) = \emptyset \tag{4.19}$$

[4] This is because we have the freedom to choose the origin of the space to be the center of the data space without loss of generality.

Furthermore, we know from Table 4.1 that

$$\begin{aligned}
\mathbb{R}^{r_t} &= \text{Identity Space } \cup \text{Variation Space} && (4.20)\\
&= (\text{Null}(\tilde{\mathbf{S}}_w) \cap \text{Range}(\tilde{\mathbf{S}}_b)) \cup (\text{Range}(\tilde{\mathbf{S}}_w) \cap \text{Null}(\tilde{\mathbf{S}}_b)) && (4.21)\\
&= \text{Range}(\tilde{\mathbf{S}}_b) \cup \text{Range}(\tilde{\mathbf{S}}_w) && (4.22)
\end{aligned}$$

Combining Equation 4.19 and 4.20, we know that

$$\begin{aligned}
\text{Identity Space} = \text{Range}(\tilde{\mathbf{S}}_b) &= \text{Null}(\tilde{\mathbf{S}}_w) && (4.23)\\
\text{Variation Space} = \text{Range}(\tilde{\mathbf{S}}_w) &= \text{Null}(\tilde{\mathbf{S}}_b) && (4.24)\\
\text{Range}(\tilde{\mathbf{S}}_w) &\perp \text{Range}(\tilde{\mathbf{S}}_b) && (4.25)
\end{aligned}$$

Theorem 4.3.6 follows directly.

Theorem 4.3.6. ***Decomposition of whitened data space*** *The whitened data space can be written as an orthogonal direct sum* [5] *of the following subspaces:*

$$\begin{aligned}
\mathbb{R}^{r_t} = Range(\tilde{\mathbf{S}}_t) &= Range(\tilde{\mathbf{S}}_b) \oplus Range(\tilde{\mathbf{S}}_w) && (4.26)\\
&= \underbrace{Range(\tilde{\mathbf{S}}_b)}_{\text{Identity Space}} \oplus \underbrace{Range(\tilde{\mathbf{S}}_{\mathbf{t}}^{\;1}) \oplus \cdots \oplus Range(\tilde{\mathbf{S}}_{\mathbf{t}}^{\;C})}_{\text{Variation Space}} && (4.27)
\end{aligned}$$

and we have

$$\begin{aligned}
Rank(Whitened\ data\ space) &= Rank(\tilde{\mathbf{S}}_t) = D - N + 1, && (4.28)\\
Rank(Identity\ Space) &= Rank(\tilde{\mathbf{S}}_b) = C - 1, && (4.29)\\
Rank(Variation\ Space) &= \sum_{i=1}^{C} Rank(\tilde{\mathbf{S}}_t^i) && (4.30)\\
&= \sum_{i=1}^{C} (N_i - 1) && (4.31)\\
&= N - C && (4.32)
\end{aligned}$$

[5] If a space S is written as the direct sum of subspaces $S_1, \ldots, S_n$, then we have $S = S_1 + \cdots + S_2$, and $S_i \bigcap S_j = \emptyset$ for $i \neq j$. The direct sum is orthogonal if $S_i \perp S_j$ for $i \neq j$.

Chapter 5

Multimodal Discriminant Analysis

The content of this chapter is built upon [23, 36, 39].

Chapter 4 has provided us with a framework of analysis to search for the most discriminating subspace for dimension reduction in the whitened data space. We have seen that the *Identity Space* is the most discriminating subspace because it only contains information regarding the class identity of a given data vector. By projecting the sample vectors to this subspace, the class information is maximally preserved.

In this chapter, we shall discuss algorithms which compute the *Identity Spaces* of multimodal data. Multimodal data are formulated as data obtained as the result of interaction of multiple factors, each of which we shall term as **mode**. For instance, face images are resultant of interaction of the person's identity, the pose of head, the lighting condition and the type of facial expression. As such, each data vector contains labeling information regarding multiple modes.

In many respects, multimodal analysis is similar to regression analysis. In regression analysis, the dependent variable is modeled as a function of the independent variables, corresponding parameters and an error term. Similarly, in multimodal analysis, a data vector is modeled as a function (in our case, as a linear combination) of several vectors, each belonging to a class under a certain mode, and a residue. The difference between them is that regression analysis usually works in the continuous domain, whereas multimodal analysis works in the discrete domain.

In this chapter, we shall assume that the given data matrix has been whitened before any analysis is carried out. We shall following the notations from Chapter 4 except that we omit the tilde for whitened matrices (the only type that we encounter in this chapter).

5.1 FFKT

We shall start with unimodal analysis in this section, i.e., with one label for each vector. The algorithm for computing the *Identity Space* in the unimodal case can be inferred directly from the analysis of Chapter 4. We shall call the algorithm the **Fisher-Fukunaga-Koontz Transform**, because of its close relationship with the whitening transform (i.e. *FKT*) and *FDA*.

5.1.1 Algorithm

The detailed algorithm is described in Algorithm 2. The most expensive operation involved is eigen-decomposition of $\mathbf{H}_b \in \mathbb{R}^{r_t \times C}$. Thus the time complexity is $O(r_t C^2)$ and the space complexity is $O(Nr_t)$ (we know that $r_t \leq N$).

Algorithm 2 Computation of projection matrices for the *Identity Space* and *Variation Space* based on *FFKT*

```
Input: Whitened data matrix $\mathbf{A} \in \mathbb{R}^{r_t \times N}$, Labeling matrix $\mathbf{L} \in \mathbb{R}^{1 \times N}$
Output: Projection matrix $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_3]$,where $\mathbf{V}_1 \in \mathbb{R}^{r_t \times r_b}$ and $\mathbf{V}_3 \in \mathbb{R}^{r_t \times (r_t - r_b)}$ are the
    eigenvector matrices of the Identity Space and Variation Space.
1: Compute $\mathbf{H}_b \in \mathbb{R}^{r_t \times C}$ of data matrix $\mathbf{A}$.
2: Compute the eigenvector matrix $\mathbf{V} = [\mathbf{V}_1\mathbf{V}_3]$ of $\mathbf{S}_b$ based on $\mathbf{H}_b$ and Remark
    3.1.1, where $\mathbf{V}_1$ corresponds to eigenvectors with eigenvalue 1 and $\mathbf{V}_3$ with
    eigenvalue 0.
```

For the convenience of notation, we shall overload the term *FFKT* with the following operation.

Definition 5.1.1. Given a *whitened* data matrix $\mathbf{A} = [a_1, \ldots, a_N] \in \mathbb{R}^{D \times N}$ and a labeling matrix, each entry denoting the label of the corresponding column vector in $\mathbf{A}$, the operation **FFKT** is defined as

$$\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_3] = \mathbf{FFKT}(\mathbf{A}, \mathbf{L}) \tag{5.1}$$

where $\mathbf{V}_1$ and $\mathbf{V}_3$ are orthogonal eigenvector matrices for the Identity and Variation Space of $\mathbf{A}$, as computed by Algorithm 2, and $[\mathbf{V}_1, \mathbf{V}_3]$ is a square orthogonal matrix [1].

5.1.2 Vector Decomposition and Representation

Once we obtain the projection matrices $\mathbf{V}_1$ and $\mathbf{V}_3$ of the *Identity Space* and the *Variation Space*, we could decompose any whitened sample vector $\tilde{\mathbf{a}} \in L_i$ in the following manner[2]:

$$\begin{aligned} \tilde{\mathbf{a}} &= \mathbf{V}_1\mathbf{V}_1^\top\tilde{\mathbf{a}} + \mathbf{V}_3\mathbf{V}_3^\top\tilde{\mathbf{a}} && (5.2) \\ &= \mathbf{V}_1\mathbf{V}_1^\top\tilde{\mathbf{a}} + \mathbf{V}_3^i(\mathbf{V}_3^i)^\top\tilde{\mathbf{a}} && (5.3) \end{aligned}$$

where $\mathbf{V}_1$, $\mathbf{V}_3$ and $\mathbf{V}_3^i$ denote the orthogonal eigenvector matrices for the Identity Space, the Variation Space and individual variation space of Class L_i (i.e. Range($\tilde{\mathbf{S}}_w^i$)) respectively.

In Equation 5.3, $\tilde{\mathbf{a}}$ is decomposed into an identity component which contains only the between-class information and a variation component which contains information regarding its variation within its class.

5.2 Recursive FFKT

This section and the next section will address the topic of Multimodal discriminant analysis, i.e., each vector is associated with several labels.

For the ease of demonstration of concepts, we shall assume that the structure of the data sample is a Cartesian product of different modes. Let's take the PIE database [24] for example. Each of the 68 persons has seven poses and photos are

[1] This is equivalent to a rotation of the coordinate system.

[2] The same remark holds for any $\mathbf{a} \in \mathbb{R}^{N-1}$, with an additional term in Equation 5.2: a projection term to the null space of the data matrix Null($\mathbf{S_t}$)

taken for each pose under 21 different lighting conditions. Thus there are 68×7×21 photos in total.

Mathematically, we consider a data sample **A**, in which each vector is associated with M labels, i.e. we have M modes of classification. We use M labeling matrices $\{\mathbf{M}_p\}_{p=1,\ldots,M}$ to represent the labels under different modes. Assuming that for each labeling mode p, there are C^p number of classes, and N^p samples per class. There is exactly one vector corresponding to each possibility of labeling $[l_1, \ldots, l_M]$, where $l_p \in \{1, \ldots, C^p\}$. Thus we have the following relationships:

$$N = \prod_{p=1}^{M} C^p \tag{5.4}$$

$$N_p = \prod_{q \neq p} C^q \tag{5.5}$$

$$N = N^p C^p \tag{5.6}$$

The central idea of the section is to perform *FFKT*[3] recursively to the whitened data space, with a different labeling for each iteration, to extract the class information for different modes step by step. However, this approach needs justification, because the extraction of the class-identity components based on different modes may be correlated, i.e. the identity component of a mode may contain identity information of another mode.

This section is structured as follows. Section 5.2.1 justifies the approach of *Recursive FFKT* regarding the issue of correlation among different identity components. Section 5.2.3 provides the mathematical formulation and detailed algorithm of *Recursive FFKT*. Finally, Section 5.2.4 provides the formula for vector decomposition and representation using the projection matrices computed by *Recursive FFKT*.

5.2.1 Beyond FFKT

We know from Section 5.1 that *FFKT* allows us to extract class-identity information of a given data sample based on its labeling information. At the presence of several modes of labeling, a natural question arises: can we extract class-identity

[3] As defined in 5.1.1

information under these different modes separately?

Firstly, let's take a closer look at *FFKT*. Suppose we perform *FFKT* on $\mathbf{A}$, based on labeling matrix $\mathbf{M}_1$, to obtain:

$$\mathbf{V}^{(1)} = [\mathbf{V}_1^{(1)}\mathbf{V}_3^{(1)}] = \text{FFKT}(\mathbf{A}^{(1)}, \mathbf{L}^{(1)}) \tag{5.7}$$

The projection of $\mathbf{A}$ to the new *FFKT*-transformed space is:

$$\mathbf{A}^{(2)} = (\mathbf{V}^{(1)})^\top \mathbf{A} = [\mathbf{V}_1^{(1)}\mathbf{V}_3^{(1)}]^\top \mathbf{A} \tag{5.8}$$

If we further apply *FFKT* to $\mathbf{A}^{(2)}$,

$$\mathbf{V}^{(2)} = [\mathbf{V}_1^{(2)}\mathbf{V}_3^{(2)}] = \text{FFKT}(\mathbf{A}^{(2)}, \mathbf{L}^{(2)}) \tag{5.9}$$

let's study the relationship between $\mathbf{V}^{(1)}$ and $\mathbf{V}^{(2)}$.

We note that the projection matrices for the extraction of class-identity component under Mode 1 and Mode 2 are respectively:

$$\mathbf{P}^{(1)} = \mathbf{V}_1^{(1)} \tag{5.10}$$

$$\mathbf{P}^{(2)} = \mathbf{V}^{(1)}\mathbf{V}_1^{(2)} \tag{5.11}$$

The multiplication term $\mathbf{V}^{(1)}$in Equation 5.11 indicates a rotation of the coordinate system, since $\mathbf{V}^{(2)}$ is computed based on $\mathbf{A}^{(2)}$, whose coordinate system is a rotation (i.e. multiplication by $\mathbf{V}^{(1)}$) of that of $\mathbf{A}$. In other words, $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$ are projection matrices defined in the space of $\mathbf{A}$, so that the extraction of class-identity components of $\mathbf{A}$ can be performed by $(\mathbf{P}^{(1)})^\top\mathbf{A}$ and $(\mathbf{P}^{(2)})^\top\mathbf{A}$ directly.

If we calculate the mean of each class under a different mode, e.g. based on labeling matrix $\mathbf{M}_2$, we have

$$\mathbf{m}_k^{(2)} = \sum_{\mathbf{a}_i \in L_k^{(2)}} \mathbf{a}_i^{(2)} \tag{5.12}$$

$$= \frac{1}{N_2} \sum_{\mathbf{a}_i \in L_k^{(2)}} [\mathbf{V}_1^{(1)} \quad \mathbf{V}_3^{(1)}]^\top \mathbf{a}_i^{(2)} \tag{5.13}$$

$$= [\frac{1}{N_2} \sum_{\mathbf{a}_i \in L_k^{(2)}} \mathbf{V}_1^{(1)} \mathbf{a}_i^{(2)} \qquad \frac{1}{N_2} \sum_{\mathbf{a}_i \in L_k^{(2)}} \mathbf{V}_3^{(1)} \mathbf{a}_i^{(2)}]^\top \tag{5.14}$$

$$= [\mathbf{0}^{1\times(C^1-1)} \qquad \frac{1}{N_2} \sum_{\mathbf{a}_i \in L_k^{(2)}} (\mathbf{V}_3^{(1)})^\top \mathbf{a}_i^{(2)}]^\top \tag{5.15}$$

Thus we have

$$\mathbf{H}_b^{(2)} = [\mathbf{m}_1^{(2)} \cdots \mathbf{m}_{C^2}^{(2)}] = \begin{bmatrix} \mathbf{0}^{(C^1-1)\times r_t} \\ \vdots \end{bmatrix} \tag{5.16}$$

then we have

$$\mathbf{V}_1^{(2)} = \begin{bmatrix} \mathbf{0}^{(C^1-1)\times r_t} \\ \vdots \end{bmatrix} \tag{5.17}$$

since $\mathbf{V}_1^{(2)}$ is the basis for the *Identity Space* under Mode 2 and we know from Equation 4.23 Section 4.3.5 that

$$\text{Identity Space } = \text{Range}(\mathbf{S}_b) = \text{Range}(\mathbf{H}_b) \tag{5.18}$$

Intuitively this means that the *Identity Space* under Mode $\mathbf{M}_2$ doesn't contain any class identity information of Mode $\mathbf{M}_1$.

Mathematically, if we want to extract the class-identity component under Mode 2, then the projection matrix is

$$\mathbf{P}^{(2)} = \mathbf{V}^{(1)}\mathbf{V}_1^{(2)} = [\mathbf{V}_1^{(1)}\mathbf{V}_3^{(1)}] \begin{bmatrix} \mathbf{0}^{(C^1-1)\times r_t} \\ \mathbf{X} \end{bmatrix} = \mathbf{V}_3^{(1)}\mathbf{X} \tag{5.19}$$

thus we have

$$\text{Range}(\mathbf{P}^{(2)}) \subseteq \text{Range}(\mathbf{V}_3^{(1)}) \tag{5.20}$$

Therefore, we have

$$\mathrm{Range}(\mathbf{P}^{(2)}) \perp \mathrm{Range}(\mathbf{P}^{(1)}) \tag{5.21}$$

since $\mathbf{P}^{(1)} = \mathbf{V}_1^{(1)}$ and $\mathbf{V}_3^{(1)} \perp \mathbf{V}_1^{(1)}$.

Equation 5.21 indicates that if we perform *FFKT* on $\mathbf{A}^{(2)}$ under the classification Mode $\mathbf{M}_2$, the extracted class-identity information is uncorrelated with the class-identity information obtained earlier under Mode $\mathbf{M}_1$, i.e. it does not contain any class-identity information of Mode $\mathbf{M}_1$. This means that we could perform *FFKT* recursively to extract *uncorrelated class-identity information* of data vectors under different modes.

5.2.2 Mathematical Formulation

Based on our analysis in Section 5.2.1, the mathematical formulation for the calculation of the projection matrix $\mathbf{V}_R^{(p)}$ for Mode $\mathbf{M}_p$ ($p = 1, \ldots, M$)under the recursive procedure is:

$$\mathbf{V}_R^{(p)} = \begin{cases} \mathrm{FFKT}(\mathbf{A}^{(p)}, \mathbf{M}_i) & \text{if } p = 1, \\ \mathrm{FFKT}\{[\mathrm{FFKT}(\mathbf{A}^{(p-1)}, \mathbf{M}_{p-1})]^\top \mathbf{A}^{(p-1)}, \mathbf{M}_p)\} & \text{if } p > 1 \end{cases} \tag{5.22}$$

5.2.3 Algorithm

Algorithm 3 provides an implementation for Equation 5.22. The time complexity is $O(Mr_tC^2)$ and the space complexity is $O(MNr_t)$ (with $r_t \leq N$).

We note that $\mathbf{V}_R^{(p)}$ is the projection matrix in the coordinate system of $\mathbf{A}^{(p)}$, which is a rotation of the original coordinate system, and needs to be rotated back if it is to be applied to the original data matrix $\mathbf{A}$. It can be shown easily from the algorithm (more precisely, Line 4 that the rotation matrix is $(\prod_{q=1}^{p-1} \mathbf{V}_R^q)$.

Therefore, the final projection matrix for identity component extraction of Mode $\mathbf{M}_p$ ($p = 1, \ldots, M$) is:

$$\mathbf{P}_R^{(p)} = \begin{cases} \mathbf{V}_{R,1}^{(p)} & \text{if } p = 1, \\ (\prod_{q=1}^{p-1} \mathbf{V}_R^q)\mathbf{V}_{R,1}^p & \text{if } p > 1 \end{cases} \tag{5.24}$$

The extraction of the identity component under Mode $\mathbf{M}_p$ is given by $(\mathbf{P}_R^{(p)})^\top \mathbf{A}$.

Algorithm 3 Computation of projection matrices for *Identity Space* and *Variation Space* for multimodal data based on *R-FFKT*

```
Input: Whitened training data matrix $\mathbf{A} \in \mathbb{R}^{r_t \times N}$, Labeling matrices $\mathbf{M}_p \in \mathbb{R}^{1 \times N}$
    ($p = 1, \dots, M$)
Output: Projection matrices $\mathbf{V}_R^p = [\mathbf{V}_{R,1}^{(p)} \mathbf{V}_{R,3}^{(p)}] \in \mathbb{R}^{r_t \times r_t}$ for extraction of identity and
    variation components under Mode $\mathbf{M}_p$ ($p = 1, \dots, M$)
Initialization: let $\mathbf{X} = \mathbf{A}$
for all $p = 1$ to $M$ do
    Compute the projection matrix of data matrix $\mathbf{X}$ with labeling matrix $\mathbf{M}_p$
        $$\mathbf{V}_R^{(p)} = [\mathbf{V}_1^{(p)} \mathbf{V}_3^{(p)}] = FFKT(\mathbf{X}, \mathbf{M}_p) \qquad (5.23)$$
    Let $\mathbf{X} = (\mathbf{V}_R^{(p)})^\top \mathbf{X}$
end for
```

5.2.4 Vector Decomposition and Representation

Given the projection matrices $\mathbf{V}_{R,1}^p \in \mathbb{R}^{(N-1) \times (C^p - 1)}$ for the *Identity Spaces* of mode p ($p = 1, \dots, M$), the data matrix can be decomposed as

$$\mathbf{X} = \sum_{p=1}^{M} \mathbf{V}_{R,1}^p (\mathbf{V}_{R,1}^p)^\top + \mathbf{V}_0 \mathbf{V}_0^\top \mathbf{X} \qquad (5.25)$$

where $\mathbf{V}_0 \in \mathbb{R}^{(N-1) \times r_0}$ is the residual space, $r_0 = N - \sum_{p=1}^{M} C^p + M - 1$.

5.3 MMDA

The objective of the analysis in this section is the same as in Section 5.2. *MMDA* seeks to extract identity information at the presence of several labeling modes. The difference from *Recursive FFKT* is: *MMDA* computes the projection matrices for different modes directly, instead of working on the rotated data samples recursively as in Section 5.2.

Similar to *Recursive FFKT*, *MMDA* requires justification since the extracted identity components for different modes may be correlated, i.e., the identity components for one mode may contain class-identity information of another. This is an undesirable situation since data samples can't be decomposed into orthogonal components, which can be analyzed independently.

The structure of the section is the following. Section 5.3.2 describes the algorithm for *MMDA* and provides a justification of the approach regarding the correlation of different identity components. Section 5.3.3 shows that *MMDA* is in fact equivalent to *Recursive FFKT* described in Section 5.2. Finally, Section 5.2.4 provides the formula for vector decomposition and representation using the projection matrices computed by *MMDA*.

5.3.1 Mathematical Formulation

We first state the mathematical formulation for the calculation of the projection matrix $\mathbf{V}_M^{(p)}$ for Mode $\mathbf{M}_p$ $(p = 1, \ldots, M)$by *MMDA*

$$\mathbf{V}_M^{(p)} = [\mathbf{V}_{M,1}^{(p)} \mathbf{V}_{M,3}^{(p)}] = FFKT(\mathbf{A}, M_p) \tag{5.26}$$

Equation 5.27 is straight forward: *MMDA* computes the projection matrices for all modes at one shot, based on the corresponding labeling matrices.

Theorem 5.3.1 justifies the approach of *MMDA* by proving that the identity spaces of different modes are orthogonal to each other, thus the extracted identity information of different modes are uncorrelated.

Theorem 5.3.1. *If $\mathbf{V}_{M,1}^{(p)}$ and $\mathbf{V}_{M,1}^{(q)}$ are projection matrices to the Identity Space for Modes $\mathbf{M}_p$ and $\mathbf{M}_q$, then*

$$(\mathbf{V}_{M,1}^{(p)})^\top \mathbf{V}_{M,1}^{(q)} = \begin{cases} \mathbf{I} & \text{if } p = q \\ \mathbf{0} & \text{if } p \neq q. \end{cases} \tag{5.27}$$

The proof is detailed in [39].

The final projection matrix identity component extraction of Mode $\mathbf{M}_p$ $(p = 1, \ldots, M)$is:

$$\mathbf{P}_M^p = \mathbf{V}_{M,1}^{(p)} \tag{5.28}$$

The extraction of the identity component under Mode $\mathbf{M}_p$ is given by $(\mathbf{P}_M^{(p)})^\top \mathbf{A}$.

5.3.2 Algorithm

Algorithm 4 provides a direct implementation of Equation 5.27. The time complexity is $O(Mr_tC^2)$ and the space complexity is $O(r_t^2)$. We note that Algorithm 4 is superior to Algorithm 3 in terms of space complexity.

Algorithm 4 Computation of projection matrices for *Identity Space* and *Variation Space* for multimodal data based on *MMDA*

```
Input: Whitened training data matrix $\mathbf{A} \in \mathbb{R}^{r_t \times N}$, Labeling matrices $\mathbf{M}_p \in \mathbb{R}^{1 \times N}$ $(p = 1, \dots, M)$
Output: Projection matrices $\mathbf{V}_R^p = [\mathbf{V}_{R,1}^{(p)} \mathbf{V}_{R,3}^{(p)}] \in \mathbb{R}^{r_t \times r_t}$ for extraction of identity and variation components under Mode $\mathbf{M}_p$ $(p = 1, \dots, M)$
for all $i = 1$ to $M$ do
    Compute the projection matrix of data matrix A with labeling matrix $M_p$
        $$\mathbf{V}_M^i = [\mathbf{V}_{M,1}^i \mathbf{V}_{M,3}^i] = FFKT(\mathbf{A}, L^i) \quad (5.29)$$
end for
```

5.3.3 Equivalence to Recursive FFKT

Both *Recursive* and *MMDA* compute projection matrices for *Identity Spaces* of different modes based on *FFKT*. It is thus natural to ask about the relationship between the outputs of the two algorithms. In this section, we are going to show that *MMDA* and *Recursive FFKT* are in fact equivalent.

Theorem 5.3.1 states that if there is a rotation of the coordinate system, the output of the *FFKT* algorithm is rotated similarly. Intuitively, this makes sense since the computation of *FFKT* is in fact based on eigen-decomposition of whitened scatter matrices.

Theorem 5.3.1. *If* $\mathbf{P}$ *is the output of FFKT for data matrix* $\mathbf{A}$ *with labeling matrix* $\mathbf{L}$, *and* $\mathbf{W}$ *is a unitary matrix (i.e. square and orthogonal), then* $\mathbf{W}^\top\mathbf{P}$ *is the output of FFKT for data matrix* $\mathbf{W}^\top\mathbf{A}$ *with the same labeling matrix* $\mathbf{L}$, *i.e.*

$$\mathbf{P} = FFKT(\mathbf{A}, \mathbf{L}) \overset{\mathbf{W}\ unitary}{\Longrightarrow} \mathbf{W}^\top\mathbf{P} = FFKT(\mathbf{W}^\top\mathbf{A}, \mathbf{L}) \quad (5.30)$$

The proof is provided in Appendix B.1.

With Theorem 5.3.1, we are ready to show the equivalence between *MMDA* and *Recursive FFKT*.

Theorem 5.3.2. *If* $\mathbf{P}_R^{(p)}$ *and* $\mathbf{P}_M^{(p)}$ *are the outputs of Recursive FFKT and MMDA for data matrix* $\mathbf{A}$ *and Mode* $\mathbf{M}_p$*, then we have:*

$$\mathbf{P}_R^{(p)} = \mathbf{P}_M^{(p)} \tag{5.31}$$

The proof is detailed in Appendix B.2.

5.3.4 Vector Decomposition and Representation

Given the projection matrices $\mathbf{V}_{M,1}^p \in \mathbb{R}^{(N-1)\times(C^p-1)}$ for the *Identity Spaces* of mode p ($p = 1, \ldots, M$), the data matrix can be decomposed as

$$\mathbf{X} = \sum_{p=1}^{M} \mathbf{V}_{M,1}^p (\mathbf{V}_{M,1}^p)^\top + \mathbf{V}_0 \mathbf{V}_0^\top \mathbf{X} \tag{5.32}$$

where $\mathbf{V}_0 \in \mathbb{R}^{(N-1)\times r_0}$ is the residual space, $r_0 = N - \sum_{p=1}^{M} C^p + M - 1$.

Chapter 6

Experiments

In this section, we test the theory of *MMDA* in two applications: face image decomposition and face recognition.

6.1 Face dataset

In our experiments, a subset of the Multi-PIE database [13] is used. The database contains 337 subjects, imaged under 15 viewpoints and 19 illumination conditions in up to four recording sessions.

Our dataset is chosen in the following manner. We first eliminate subjects who are not present in all the four sessions and who wear glasses. There are 64 subjects who remain. Then we use photos of these 64 subjects recorded during Session 02. There are 19 illumination conditions and three expressions (neutral, surprise, squint) in Session 02. Thus there are in total $64 \times 19 \times 3$ photos in our dataset.

6.2 Face Recognition

In this section, we carry out face recognition experiments. This is done by projecting probe images to the *Identity Space* of the training data labeled according to their *person identity* (computed using Algorithm 4 on Page 49). Similar to Section 6.3, we shall investigate the power of *MMDA* in different scenarios: two modes and three modes.

6.2.1 Recognition across Illumination

In this experiment, we perform face recognition across illumination. Our dataset consists of the 64 subjects with neutral expression taken under 19 illuminations. There are in total 64×19 images.

In face recognition, the small-sample-size problem is common. We thus evaluate the performance under such situations. We randomly choose n training samples from each subject, $n = 2, 4, 6, 8, 10, 12$, and the remaining samples are used for testing. Moreover, for each set of the n training samples, we repeat sampling for 10 times to compute the mean recognition rate. We compare *MMDA* with two popular methods in face recognition: *PCA* and *LDA*[1].

Figure 6.1 shows that *MMDA* consistently outperforms *PCA* and *LDA* and the overall accuracies increase with the number of training samples from each class.

6.2.2 Recognition across Expression

In this experiment, we perform face recognition across expression. Our dataset consists of the 64 subjects taken without flash and with three expressions: neutral, surprise and squint. There are in total 64×3 images.

Similar to Section 6.2.1, we randomly choose n training samples from each subject, $n = 1, 2$, and the remaining samples are used for testing. Moreover, for each set of the n training samples, we repeat sampling for 10 times to compute the mean recognition rate. We compare *MMDA*, *PCA* and *LDA*.

Figure 6.2 shows that *MMDA* still consistently outperforms *PCA* and *LDA* in this setting, although the overall accuracies degrade compared to Section 6.2.1. This may be due to the fewer training images in this setting and that the expression variation is harder to capture than that of illumination since it might not be linear.

6.2.3 Recognition across Illumination and Expression

In this experiment, we perform face recognition across illumination and expression. Our dataset consists of the 64 subjects with three expressions (neutral, surprise and

[1] Our implementation of *LDA* is based on the Regularized LDA discussed in [19]. On our dataset, this implementation has shown to be superior to *Fisherface*

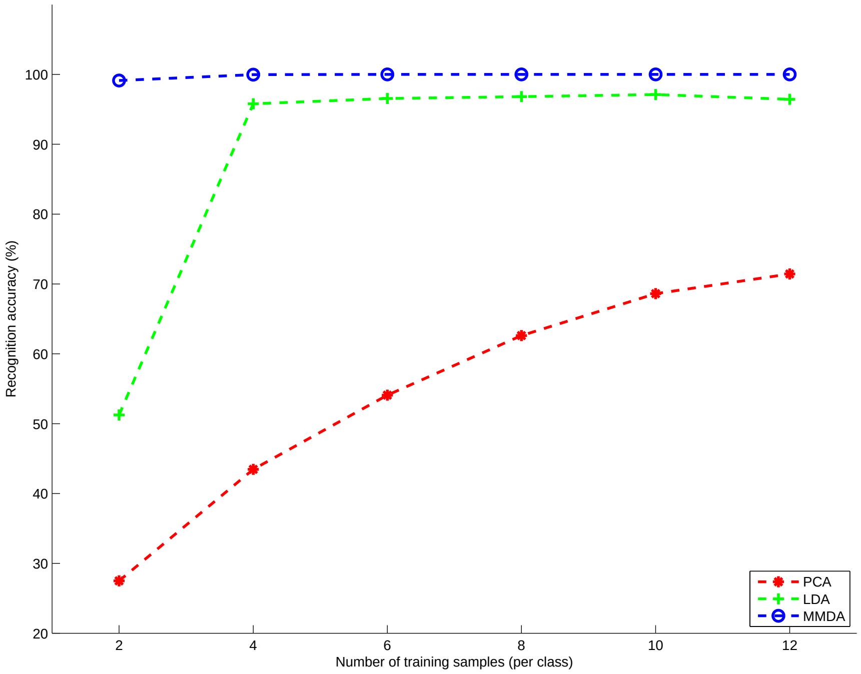


Figure 6.1: Accuracy curves for recognition across illumination on Multi-PIE images with varying training samples. The accuracy shown is the mean rate computed from 10 runs.

squint) taken under 19 illuminations. There are in total $64 \times 19 \times 3$ images.

Similar to Section 6.2.1 and 6.2.2, we randomly choose n training samples from each subject, $n = 5, 10, 15, 20, 25, 30$, and the remaining samples are used for testing. Moreover, for each set of the n training samples, we repeat sampling for 10 times to compute the mean recognition rate. We compare *MMDA* with *PCA* and *LDA*[2].

Figure 6.1 shows that *MMDA* consistently outperforms *PCA* and *LDA* and the overall accuracies increase with the number of training samples from each class.

Experiment results as demonstrated in Figure 6.1, 6.2, and 6.3 show that at the presence of illumination and/or expression variation, the discriminating power of the *Identity Space* of data labeled according to their *person identity* is significantly superior to that of the subspaces used by *LDA* or *PCA*. This is consistent to the finding

[2] Our implementation of *LDA* is based on the *Regularized LDA* discuss in [18, 19]. On our dataset, this implementation has shown to be superior than *Fisherface*.

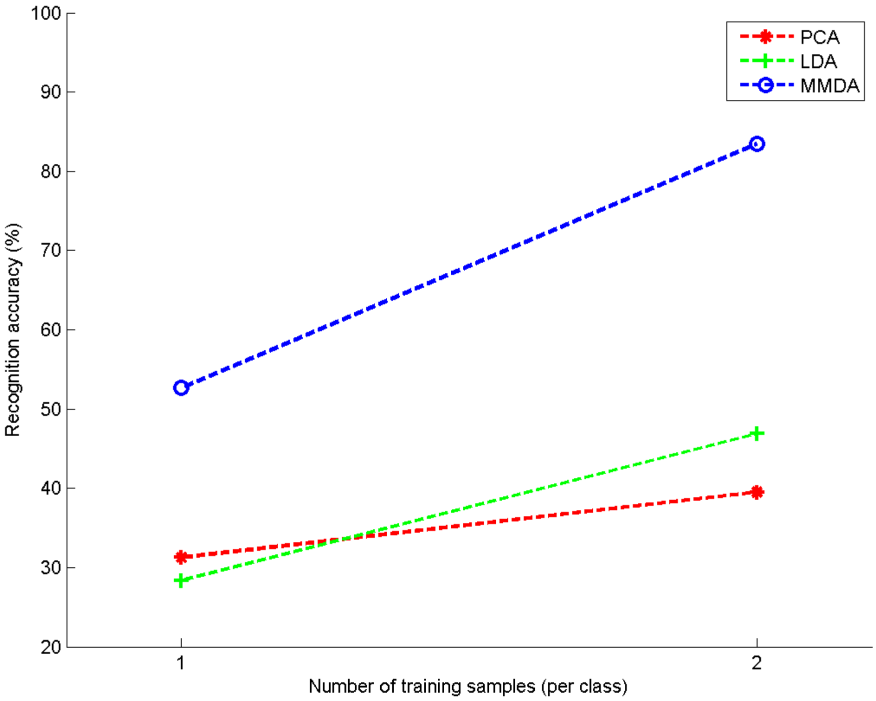


Figure 6.2: Accuracy curves for recognition across expression on Multi-PIE images with varying training samples. The accuracy shown is the mean rate computed from 10 runs.

in [35], although they only performed recognition experiments across illumination.

6.3 Factor Component Extraction and Reconstruction of Face Images

In this experiment, we implement Algorithm 4 on Page 49 to extract the person identity, illumination and/or expression information of face images.

More precisely, we perform the following procedures as outlined in Section 3.2.4:

- Whiten the dataset using Equation 3.20;
- Use Algorithm 4 to obtain the projection matrices to *Identity Spaces* of different

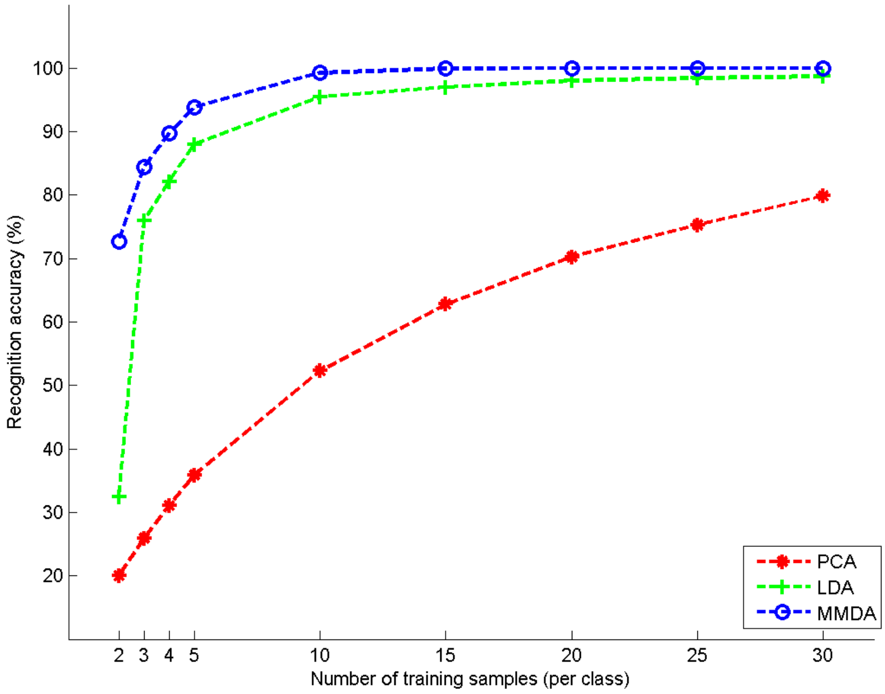


Figure 6.3: Accuracy curves for recognition across illumination and expression on Multi-PIE images with varying training samples. The accuracy shown is the mean rate computed from 10 runs.

modes in the whitened data space;

- Perform component extraction and reconstruction in the whitened domain;
- Reverse-whiten the data using Equation 3.21.

Similar to Section 6.2, we shall compare our method with *PCA*. However, we shall omit comparison with *LDA*. This is because *LDA* is rarely used for dimension reduction and reconstruction purposes, although this is theoretically possible since it only involves linear orthogonal projections. (*PCA*, on the other hand, has become a benchmark in the face recognition literature, although it was initially not designed for the application.)

6.3.1 Two Modes

We first train our *MMDA* model with two modes: person identity and illumination. For this purpose, we use a dataset consisting of 10 subjects under 14 illumination conditions with neutral expression.

We first find the projection matrices for *Identity Spaces* of Mode *Person Identity* and *Illumination*. In order to visualize the *Identity Spaces*, we show, in Figure 6.4 and 6.5, the *Identity Vectors* defined in Section 4.3.2 under the two modes.

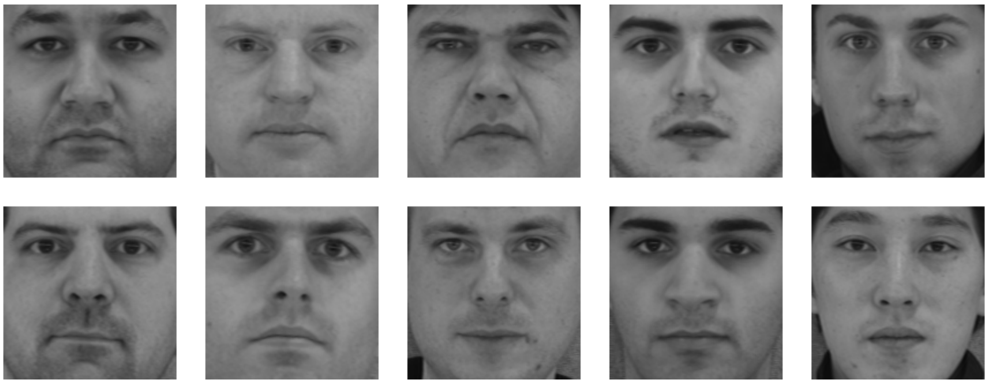


Figure 6.4: Identity Vectors for Mode *Person Identity* (The training dataset contains two modes: *Person Identity* and *Illumination*)

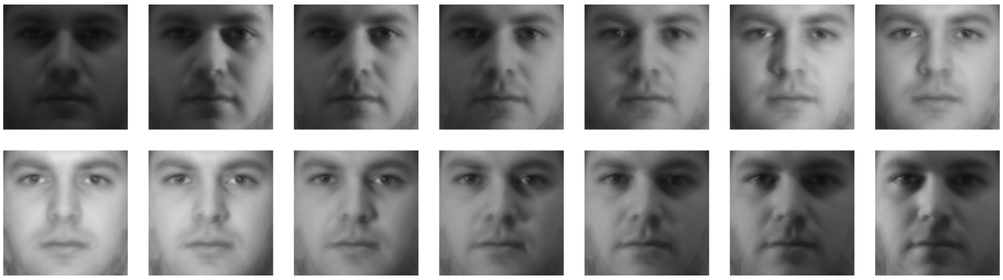


Figure 6.5: Identity Vectors for Mode *Illumination* (The training dataset contains two modes: *Person Identity* and *Illumination*)

In Figure 6.9, we show four examples of the extraction of *Person Identity* and *Illumination* components using *MMDA*. The second and third columns are the extracted components, and then the reconstructed images by *MMDA* (using Equation 5.32 without the residual term) and *PCA* (using the same number of coefficients) are shown on the fourth and fifth columns.

We note that in this setting, although the RMSE of *MMDA*-reconstructed image is larger than that of *PCA*-reconstructed image, the visual qualities are of no significant difference. In addition, *MMDA* achieves a clean and accurate decomposition of the images into two variation components.

6.3.2 Three Modes

In this section, we train our *MMDA* model with three modes: person identity, illumination and expression. For this purpose, we use a dataset consisting of the same 10 subjects under 14 illumination conditions, with three expressions: neutral, surprise and squint.

Similar to Section 6.3.1, We find the projection matrices for *Identity Spaces* of Mode *Person Identity*, *Illumination* and *Expression*. We visualize the *Identity Spaces* by showing the *Identity Vectors* under the three modes:

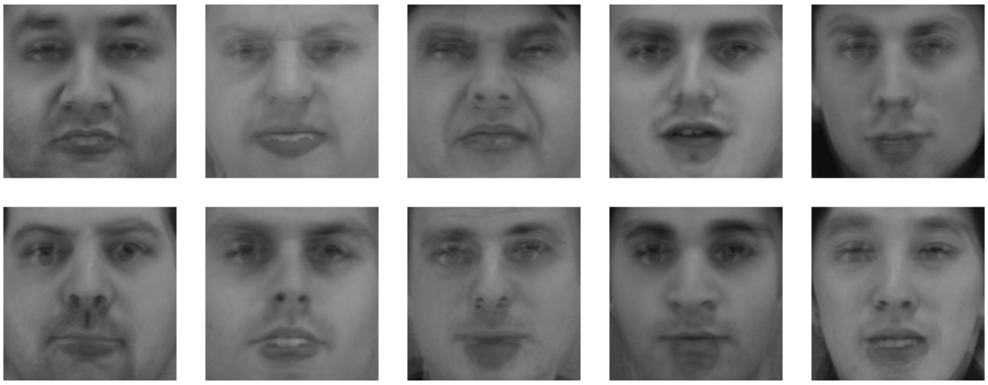


Figure 6.6: Identity Vectors for Mode *Person Identity* (The training dataset contains three modes: *Person Identity*, *Illumination* and *Expression*)

Again, in Figure 6.10, we show four examples of the extraction of *Person Identity*, *Illumination* and *Expression* components using *MMDA*. The second, third and fourth columns are the extracted components, and then the reconstructed images by *MMDA* (using Equation 5.32 without the residual term) and *PCA* (using the same number of coefficients) are shown on the fifth and sixth columns.

We note that both the extracted variation component of *person identity* and the reconstructed image using *MMDA* degrades compared to Figure 6.9 in terms of visual quality. In our setting, this is mainly due to the *discrete nature* of expression

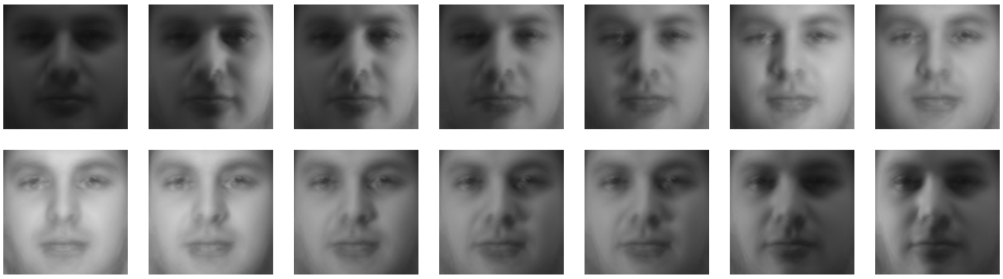


Figure 6.7: Identity Vectors for Mode *Illumination* (The training dataset contains three modes: *Person Identity*, *Illumination* and *Expression*)

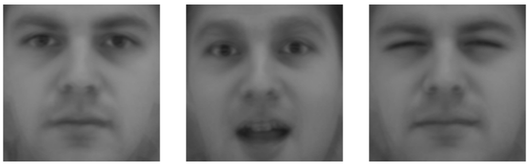
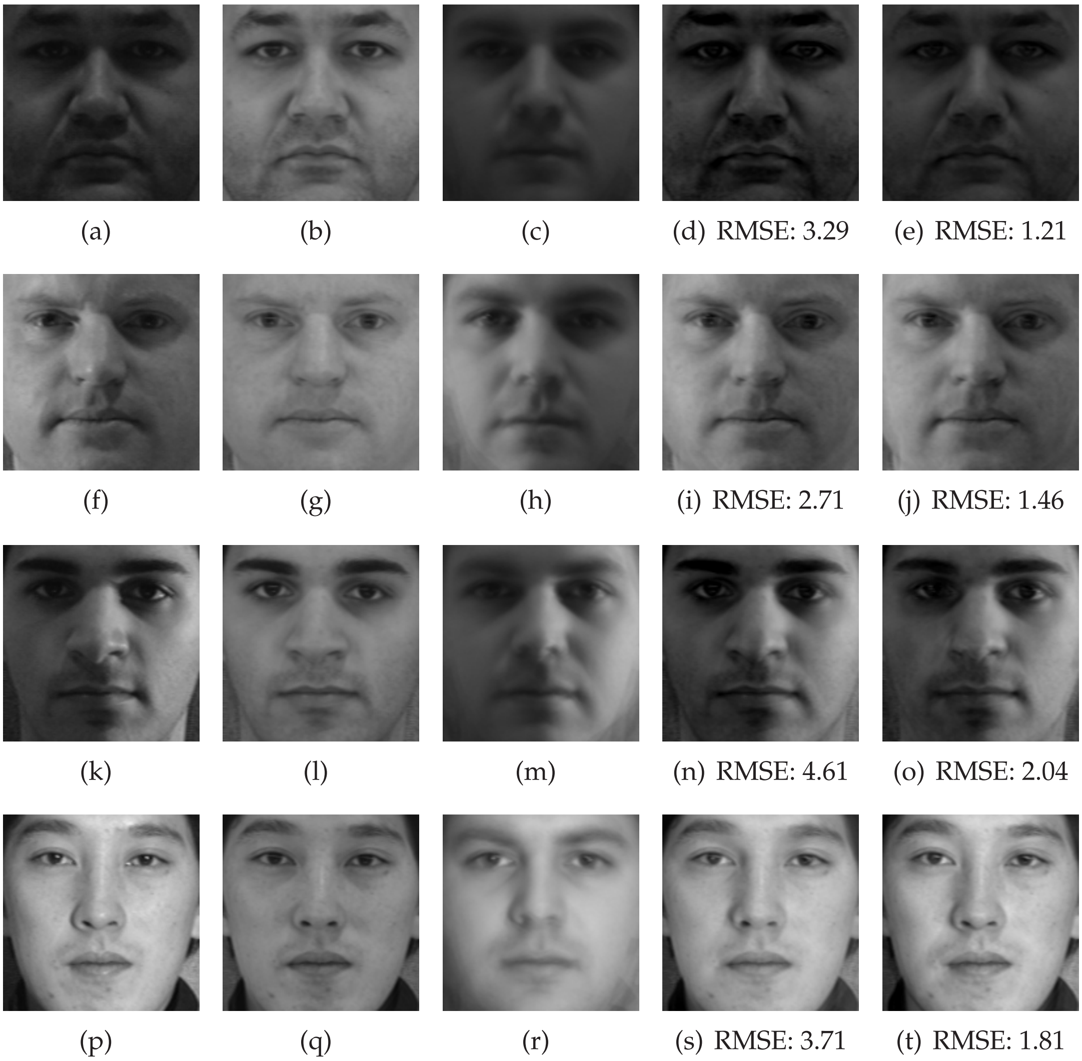


Figure 6.8: Identity Vectors for Mode *Expression* (The training dataset contains three modes: *Person Identity*, *Illumination* and *Expression*)

variation. For example, we have no information regarding the transition between a neutral expression and a surprise. This is unlike the illumination variation, of which the change is rather smooth. We suspect that if we could have a video recording of the transitions between different expressions of the subjects, the decomposition results would be improved significantly.



(a) (b) (c) (d) RMSE: 3.29 (e) RMSE: 1.21

(f) (g) (h) (i) RMSE: 2.71 (j) RMSE: 1.46

(k) (l) (m) (n) RMSE: 4.61 (o) RMSE: 2.04

(p) (q) (r) (s) RMSE: 3.71 (t) RMSE: 1.81

Figure 6.9: Two-mode scenario: Illustration of the extraction of variation components and the reconstruction of face images. 1st column: original images. 2nd column: person identity component. 3rd column: illumination component. 4th column: reconstructed image using components in the 2nd and 3rd columns. 5th column: reconstructed image using PCA with the same number of coefficients as in the 4th column.

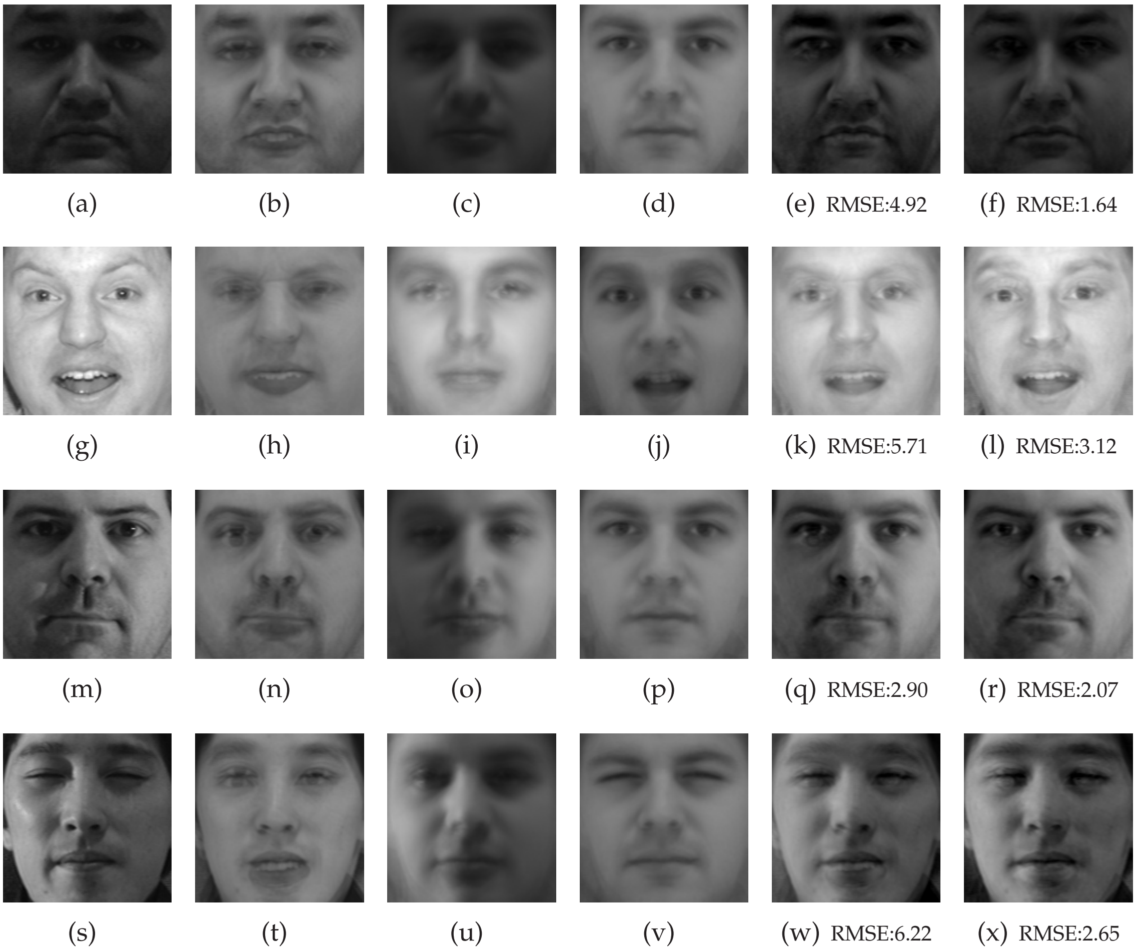


Figure 6.10: Three-mode scenario: Illustration of the extraction of variation components and the reconstruction of face images. 1st column: original images. 2nd column: person identity component. 3rd column: illumination component. 4th column: expression component. 5th column: reconstructed image using components in the 2nd, 3rd and 4th columns. 6th column: reconstructed image using PCA with the same number of coefficients as in the 5th column.

Chapter 7

Conclusion

7.1 Summary

The essential idea of the thesis is to use the most *discriminant* subspace of a certain mode (e.g. person identity, illumination or expression) as the most *representative* subspace. This is justified by the fact that the data has been whitened before the analysis is carried out, so that all the axes are weighted equally and all information regarding the representation of a certain mode comes from the labels. We have provided direct answers to the three issues raised in Section 1.1:

- We have identified the component of a face image corresponding to a certain variation mode as the *Identity Space* of the corresponding mode in the whitened data space, as described in Chapter 4;
- We can extract these components separately by performing *MMDA* in the whitened data space, as described in Chapter 5;
- The extracted components are uncorrelated[1] in the whitened space, as described in Chapter 5. However, their reconstruction in the original data space (after reverse-whitening) may still be correlated, as described in 3.2.5. This indicates that we have in fact uncovered an *uncorrelated embedding* (in the whitened space) of the inherently correlated variation components.

[1] If the data distribution follows a Gaussian distribution, uncorrelatedness is equivalent to independence. Otherwise, it is a less stronger argument since it only imposes constraints on second-order statistics.

More precisely, the extraction of the face subspace involves the following two steps:

- Whiten the data to decorrelate and normalize the axes, as described in Section 3.2;
- Find the *Identity Spaces*, i.e., the most discriminating subspaces, for each mode based on different labeling information, and the *identity vectors* in the whitened space, as described in Chapter 5

We note that the subspace obtained in this manner lies, in fact, in the whitened data space. If we wish to extract the component of an image related to a certain mode, the procedures are the following:

- Whiten the image
- Compute the projection of the whitened image to the *Identity Space* of the mode
- Reverse-whiten the projected image to the original space

We have tested the effectiveness of the framework in two applications: factor extraction and reconstruction of face images, and face recognition. We have seen that in the first application, *MMDA* does a great job (in terms of visual quality) at the presence of two modes (person identity and illumination), but degrades significantly when an additional mode (expression) is present. Very possibly this is due to the *discreet* nature of the expression data available in our dataset (i.e. we have no information regarding the transition between any two expressions). In the second application, *MMDA* outperforms traditional methods (*PCA* and *LDA*) in all cases, which verifies our claim that the *Identity Space* is the most discriminating subspace (for the mode of person identity).

At the same time, the method also has several constraints:

- It assumes a linear representation of the face image space and it relies on second-order statistics;

- Its generalization ability is limited, as with *PCA* or *LDA*, and the effectiveness of the model in terms of representation and classification power is intimately related to the nature of the training data with respect to the underlying data space.

7.2 Contributions

This thesis has the following contributions:

- Consolidates and provides a unifying formulation for previous (published and unpublished) works on *FKT*, *FFKT* and *MMDA* in [35],[36], [37], [38] and [39];
- Proves the equivalence of two multimodal extensions of *FFKT*: *Recursive FFKT* and *MMDA* put forward in [23] and [39] respectively.
- Empirically validates the algorithm of *MMDA* in the applications of face recognition and dimension reduction.

7.3 Future Directions

The focus of this thesis has been rather theoretical and work needs to be done to improve the applicability of the theory. We propose the following for future research directions:

- Use video recordings which show smooth facial expression transitions to better learn the subspace of facial expressions;
- Apply the method to face image synthesis, such as the transfer of facial expressions;
- Apply the method to types of data other than face images (for example, music recordings), since the method proposed in this thesis is potentially applicable to any multi-modal data with high dimensions.

Bibliography

[1] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, **12**:2385–2404, Dec 2000.

[2] Peter N. Belhumeur, Joao P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(7):711–720, Jul 1997.

[3] Volker Blanz, Patrick Grother, P. Jonathon Phillips, and Thomas Vetter. Face recognition based on frontal views generated from non-frontal images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **2**, pages 454–461, Jun 2005.

[4] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of ACM SIGGRAPH*, pages 187–194, 1999.

[5] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**(9):1063–1074, Sep 2003.

[6] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. In *Proceedings of the European Conference on Computer Vision*, **2**, pages 484–498, 1998.

[7] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**(6):681–685, Jun 2001.

[8] Nicholas P. Costen, Tim F. Cootes, Gareth J. Edwards, and Chris J. Taylor. Simultaneous extraction of functional face subspaces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **1**, pages 492–497, 1999.

[9] Trevor F. Cox and Michael A.A. Cox. *Multidimensional Scaling*. Chapman & Hall, 1994.

[10] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, second edition, Nov 2000.

[11] Stephen H. Friedberg, Arnold J. Insel, and Lawrence E. Spence. *Linear Algebra*. Prentice Hall, second edition, 1989.

[12] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, New York, second edition, 1990.

[13] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, Sep 2008.

[14] Dong Guo. Face recognition. Graduate Research Paper, 2006 Oct.

[15] Jian Huang, Pong C. Yuen, Wen-Sheng Chen, and Jian Huang Lai. Choosing paramters of kernel subspace lda for recognition of face images under pose and illumination variations. *IEEE Transactions on Systems, Man, and Cybernetics*, **37**(4):847–862, Aug 2007.

[16] Rui Huang, Qingshan Liu, Hanqing Lu, and Songde Ma. Solving the small sample size problem of lda. In *Proceedings of the IEEE International Conference on Pattern Recognition*, **3**, pages 29 – 32, Aug 2002.

[17] Xiaoming Huo. A statistical analysis of fukunaga-koontz transform. *IEEE Signal Processing Letters*, **11**:123–126, 2004.

[18] Juwei Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Regularization studies on lda for face recognition. In *Proceedings of IEEE International Conference on Image Processing*, **1**, pages 63–66, Oct 2004.

[19] Juwei Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition. *Pattern Recognition Letters*, **26**(2):181–191, Jan 2005.

[20] Juwei Lu, Konstantinos N. Plataniotis, and Anastasios .N. Venetsanopoulos. Face recognition using kernel direct discriminant analysis algorithms. *IEEE Transactions on Neural Networks*, **14**(1):117–126, Jan 2003.

[21] Aleix .M. Martínez and Avinash C. Kak. Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**:228–233, Feb 2001.

[22] Bernhard Scholkopf, Alexander Smola, and Klaus-Robert Muller. Nonlinear component analysis as a kernel eigenvalue problem. Technical Report 44, Max-Planck-Institut, Germany, Dec 1996.

[23] Terence Sim. Recursive ffkt. Unpublished notes, 2008.

[24] Terence Sim, Simon Baker, and Maan Bsat. The cmu pose, illumination, and expression (pie) database. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.

[25] Terence Sim and Sheng Zhang. Exploring face space. In *Proceedings of IEEE Computer Science Conference on Computer Vision and Pattern Recognition Workshops*, Jun 2004.

[26] Joshua B. Tenenbaum, Vin de Silva, and John C.Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, **290**:2319–2333, Dec 2000.

[27] Matthew A. Turk and Alex P. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, **3**(1):71–86, 1991.

[28] Matthew A. Turk and Alex P. Pentland. Face recognition using eigenfaces. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[29] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *Proceedings of the European Conference on Computer Vision*, pages 447–460, May 2002.

[30] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear image analysis for facial recognition. In *Proceedings of the International Conference on Pattern Recognition*, **2**, pages II: 511–514, 2002.

[31] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear subspace analysis of image ensembles. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **2**, pages II: 93–99, Jun 2003.

[32] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. In *Proceedings of ACM SIGGRAPH*, 2005.

[33] Ming-Hsuan Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.

[34] Sheng Zhang. *Exploring Face Space: A Computational Approach*. PhD thesis, National University of Singapore, 2006.

[35] Sheng Zhang and Terence Sim. When fisher meets fukunaga-koontz: A new look at linear discriminants. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[36] Sheng Zhang and Terence Sim. Discriminant subspace analysis: A fukunaga-koontz approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**(10):1732 – 1745, Oct 2007.

[37] Sheng Zhang and Terence Sim. Identity space: Where classes are perfectly seperated. Unpublished manuscript, 2007.

[38] Sheng Zhang and Terence Sim. Variation space: No means for classification. Unpublished manuscript, 2007.

[39] Sheng Zhang and Terence Sim. Multimodal discriminant analysis. Unpublished manuscript, 2008.

[40] Shaohua Kevin Zhou, Rama Chellappa, and Baback Moghaddam. Intrapersonal kernel space for face recognition. In *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pages 235–240, May 2004.

Appendix A

Rank of Scatter Matrices and Precursor Matrices

Theorem A.1. ***Rank-Nullity Theorem*** *The rank and the nullity of a matrix add up to the number of columns of the matrix. Specifically, if* $\mathbf{A}$ *is an* $m \times n$ *matrix, then* $Rank(\mathbf{A}) + Nullity(\mathbf{A}) = n$.

The proof of this theorem can be found in any linear algebra textbook such as [11].

Lemma A.2. *For any given matrix* $\mathbf{A}$*, we have*

$$Rank(\mathbf{A}^\top\mathbf{A}) = Rank(\mathbf{A}\mathbf{A}^\top) = Rank(\mathbf{A}) \tag{A.1}$$

Proof. Secondly, we have

$$\begin{aligned} & \mathbf{A}^\top\mathbf{A}\mathbf{x} = \mathbf{0} \\ \Rightarrow \quad & \mathbf{x}\mathbf{A}^\top\mathbf{A}\mathbf{x} = |\mathbf{A}\mathbf{x}|^2 = |\mathbf{A}\mathbf{x}| = 0 \\ \Rightarrow \quad & \mathbf{A}\mathbf{x} = \mathbf{0} \end{aligned} \tag{A.2}$$

and

$$\begin{aligned} & \mathbf{A}\mathbf{x} = 0 \\ \Rightarrow \quad & \mathbf{A}^\top\mathbf{A}\mathbf{x} = \mathbf{0} \end{aligned} \tag{A.3}$$

Equations A.2 and A.3 give us

$$\mathbf{Ax} = 0 \quad \Leftrightarrow \quad \mathbf{A}^\top\mathbf{Ax} = \mathbf{0} \tag{A.4}$$

which means the nullity of $\mathbf{A}$ is equal to the nullity $\mathbf{A}^\top\mathbf{A}$.

Moreover, we notice they have the same number of columns. Thus we have

$$\text{Rank}(\mathbf{A}) = \text{Rank}(\mathbf{A}^\top\mathbf{A}) \tag{A.5}$$

Similarly, we have

$$\text{Rank}(\mathbf{A}^\top) = \text{Rank}(\mathbf{A}\mathbf{A}^\top) \tag{A.6}$$

We further notice that [11]

$$\text{Rank}(\mathbf{A}) = \text{Rank}(\mathbf{A}^\top) \tag{A.7}$$

Combining Equations A.5, A.6, A.7, and Theorem A.1 the statement is proved. □

Lemma A.3. *For any matrix* **A** *and full rank matrix* **U**[1], *we have*

$$Rank(\mathbf{U}^\top\mathbf{A}) = Rank(\mathbf{A}) \tag{A.8}$$

Proof. Since **U** is of full rank [11], we have

$$\begin{aligned} &\mathbf{U}^\top\mathbf{Ax} = \mathbf{0} \\ \Rightarrow \quad &\mathbf{Ax} = \mathbf{0} \end{aligned} \tag{A.9}$$

Obviously we also have

$$\begin{aligned} &\mathbf{Ax} = \mathbf{0} \\ \Rightarrow &\mathbf{U}^\top\mathbf{Ax} = \mathbf{0} \end{aligned} \tag{A.10}$$

Thus $\mathbf{U}^\top\mathbf{A}$ and $\mathbf{A}$ have the same null space.

Moreover, they have the same number of columns.

[1] A $m \times n$ matrix is of full rank if its rank is equal to $min(m, n)$

Therefore, using Theorem A.1, the statement is proved. □

Theorem A.4. *Let* $\mathbf{S}$ *be any of the three scatter matrices* $\mathbf{S} \in \{\mathbf{S}_t, \mathbf{S}_w, \mathbf{S}_t\}$, $\mathbf{H}$ *be the corresponding precursor matrix* $\mathbf{H} \in \{\mathbf{H}_t, \mathbf{H}_w, \mathbf{H}_t\}$, $\tilde{\mathbf{S}}$ *be the corresponding whitened scatter matrix scatter matrices* $\tilde{\mathbf{S}} \in \{\tilde{\mathbf{S}}_t, \tilde{\mathbf{S}}_w, \tilde{\mathbf{S}}_t\}$, *and* $\tilde{\mathbf{H}}$ *be the corresponding whitened precursor matrices* $\{\tilde{\mathbf{H}}_t, \tilde{\mathbf{H}}_w, \tilde{\mathbf{H}}_t\}$. *Their ranks are equal, i.e.,*

$$Rank(\mathbf{S}) = Rank(\mathbf{H}) = Rank(\tilde{\mathbf{S}}) = Rank(\tilde{\mathbf{H}}) \tag{A.11}$$

Proof. Since $\mathbf{S} = \mathbf{H}\mathbf{H}^\top$ and $\tilde{\mathbf{S}} = \tilde{\mathbf{H}}\tilde{\mathbf{H}}^\top$, we have

$$\text{Rank}(\mathbf{S}) = \text{Rank}(\mathbf{H}) \tag{A.12}$$

$$\text{Rank}(\tilde{\mathbf{S}}) = \text{Rank}(\tilde{\mathbf{H}}) \tag{A.13}$$

based on Lemma A.2.

Since $\tilde{\mathbf{H}} = \mathbf{P}^\top\mathbf{H}$, where $\mathbf{P}$ [2] is of full rank ($\mathbf{P} \in \mathbb{R}^{D\times r_t}$ and $\text{Rank}(\mathbf{P}) = r_t$), we have

$$\text{Rank}(\mathbf{H}) = \text{Rank}(\tilde{\mathbf{H}}) \tag{A.14}$$

based on Lemma A.3.

Combining Equations A.12, A.13 and A.14, the statement is proved. □

[2] Defined in Section 3.2.

Appendix B

Equivalence of Recursive FFKT and MMDA

B.1 Proof of Theorem 5.3.1

First, let's start with a lemma.

Lemma B.1.1. *If* $\mathbf{U}$ *is an eigenvector matrix for* $\mathbf{A}$*, and* $\mathbf{V}$ *is a unitary matrix (i.e. square and orthogonal), then* $\mathbf{V}^\top\mathbf{U}$ *is an eigenvector matrix for* $\mathbf{V}^\top\mathbf{A}\mathbf{V}$*.*

Proof. If $\mathbf{D}$ is the similarity matrix of $\mathbf{A}$ corresponding to the eigenvector matrix $\mathbf{U}$, i.e.

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^\top \tag{B.1}$$

then we have

$$\mathbf{V}^\top\mathbf{A}\mathbf{V} = (\mathbf{V}^\top\mathbf{U})\mathbf{D}(\mathbf{V}^\top\mathbf{U})^\top \tag{B.2}$$

The statement follows directly from Equation B.2.

□

Proof. In order to prove the statement, let's recall the algorithm of FFKT on Page 41. We shall denote the variables encountered during the computation on the right hand side of Equation 5.30 with a bar on top. For example, let's denote $\bar{\mathbf{A}} = \mathbf{W}^\top\mathbf{A}$.

Firstly, the whitening transforms $\mathbf{Q}$ and $\bar{\mathbf{Q}}$ are computed based on the total scatter matrix $\mathbf{S_t} = \mathbf{A}\mathbf{A}^\top$ and $\bar{\mathbf{S}}_t = \bar{\mathbf{A}}\bar{\mathbf{A}}^\top = \mathbf{W}^\top\mathbf{S}_t\mathbf{W}$. Based on Lemma B.1.1, we know that $\bar{\mathbf{Q}} = \mathbf{W}^\top\mathbf{Q}$.

Secondly, the whitening transform is applied to the between-class scatter matrices:

$$\tilde{\mathbf{S}}_b = \mathbf{Q}^\top\mathbf{S}_b\mathbf{Q} \tag{B.3}$$

and

$$\tilde{\bar{\mathbf{S}}}_b = \bar{\mathbf{Q}}^\top\bar{\mathbf{S}}_b\bar{\mathbf{Q}} = (\mathbf{W}^\top\mathbf{Q})^\top(\mathbf{W}^\top\mathbf{S}_b\mathbf{W})(\mathbf{W}^\top\mathbf{Q}) = \mathbf{Q}^\top\mathbf{S}_b\mathbf{Q} = \tilde{\mathbf{S}}_b \tag{B.4}$$

Thirdly, the projection matrices $\mathbf{V}$ and $\bar{\mathbf{V}}$ to the *Identity Space* and *Variation Space* are computed as the eigenvector matrices corresponding to the unit and zero eigenvalues of $\tilde{\mathbf{S}}_b$ and $\tilde{\bar{\mathbf{S}}}_b$. We know from the previous step that $\tilde{\mathbf{S}}_b = \tilde{\bar{\mathbf{S}}}_b$, thus we have $\mathbf{V} = \bar{\mathbf{V}}$.

Finally, the final projection matrices are computed as $\mathbf{P} = \mathbf{Q}\mathbf{V}$ and $\bar{\mathbf{P}} = \bar{\mathbf{Q}}p\bar{\mathbf{V}} = \mathbf{W}^\top\mathbf{Q}\mathbf{V}$. Therefore we have $\bar{\mathbf{P}} = \mathbf{W}^\top\mathbf{P}$. □

B.2 Proof of Theorem 5.3.2

Proof. We recall from Section 5.2.3 and 5.3.2 that

$$\mathbf{P}_R^{(p)} = \begin{cases} \mathbf{V}_{R,1}^{(p)} & \text{if } p = 1, \\ (\prod_{q=1}^{p-1} \mathbf{V}_R^q)\mathbf{V}_{R,1}^p & \text{if } p > 1 \end{cases} \tag{B.5}$$

$$\mathbf{P}_M^p = \mathbf{V}_{M,1}^{(p)} \tag{B.6}$$

Let's consider two cases.

- **p = 1**

  In this case, we have

$$\mathbf{V}_R^{(p)} = [\mathbf{V}_{R,1}^{(p)}\mathbf{V}_{R,3}^{(p)}] = \text{FFKT}(\mathbf{A}, \mathbf{M}_p) \tag{B.7}$$

$$\mathbf{V}_M^{(p)} = [\mathbf{V}_{M,1}^{(p)}\mathbf{V}_{M,3}^{(p)}] = \text{FFKT}(\mathbf{A}, \mathbf{M}_p) \tag{B.8}$$

thus obviously

$$\mathbf{P}_R^{(p)} = \mathbf{P}_M^{(p)} \tag{B.9}$$

- **p > 1**

We see from Algorithm 3 that, after iteration p, the data matrix $\mathbf{X}$ is computed as

$$\begin{aligned} \mathbf{X} &= (\mathbf{V}_R^{(1)})^\top \cdots (\mathbf{V}_R^{(p)})^\top \mathbf{A} && \text{(B.10)} \\ &= (\mathbf{V}_R^{(1)} \cdots \mathbf{V}_R^{(p)})^\top \mathbf{A} && \text{(B.11)} \end{aligned}$$

Since $\forall p = 1, \ldots, M$, $\mathbf{V}_R^{(p)}$ is a unitary matrix, their product is also a unitary matrix. Thus Theorem 5.3.1 applies and we have

$$\begin{aligned} \mathbf{V}_R^{(p)} &= FFKT(\mathbf{X}, \mathbf{M}_p) && \text{(B.12)} \\ &= FFKT((\mathbf{V}_R^{(1)} \cdots \mathbf{V}^{(p-1)})^\top \mathbf{A}, \mathbf{M}_p) && \text{(B.13)} \\ &= (\mathbf{V}_R^{(1)} \cdots \mathbf{V}^{(p-1)})^\top FFKT(\mathbf{A}, \mathbf{M}_p) && \text{(B.14)} \\ &= (\mathbf{V}_R^{(1)} \cdots \mathbf{V}_R^{(p-1)})^\top \mathbf{V}_M^{(p)} && \text{(B.15)} \end{aligned}$$

which leads to

$$\mathbf{V}_M^{(p)} = \mathbf{V}_R^{(1)} \cdots \mathbf{V}_R^{(p-1)} \mathbf{V}_R^{(p)} \tag{B.16}$$

and thus we have

$$\mathbf{V}_{M,1}^{(p)} = \mathbf{V}_R^{(1)} \cdots \mathbf{V}_R^{(p-1)} \mathbf{V}_{R,1}^{(p)} \tag{B.17}$$

(since $\mathbf{V}_{M,1}^{(p)}$ and $\mathbf{V}_{R,1}^{(p)}$ are the first $C^p - 1$ columns of $\mathbf{V}_M^{(p)}$ and $\mathbf{V}_R^{(p)}$ respectively) Combining Equation B.5, B.6 and B.17, we have for $p > 1$,

$$\mathbf{P}_R^{(p)} = \mathbf{P}_M^{(p)} \tag{B.18}$$

□